Rollover Prevention and Path Following of a Scaled Autonomous Vehicle Using Nonlinear Model Predictive Control

By:
Isaac Gwayi

Submitted to the Department of Electrical Engineering,
University of Cape Town, in fulfillment of the requirements for a Degree of Master of Science in Engineering

Supervised by:

Dr. M.S. Tsoeu
Dept. of Electrical Engineering
University of Cape Town

5th November, 2018

Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this report from the work(s) of other people has been attributed, and has been cited and referenced.
3. This report is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work or part thereof.

Isaac Gwayi

Signature Signed by candidate

Date: 5th November, 2018.

Publication

Part of the work in this report was published.

I. Gwayi and M.S. Tsoeu, “Rollover prevention and path following of autonomous vehicle using nonlinear model predictive control,” in *IEEE Open Innovations Conference*, Johannesburg, October 2018 (In Press).

Abstract

Vehicle safety remains an important topic in the automotive industry due to the large number of vehicle accidents each year. One of the causes of vehicle accidents is due to vehicle instability phenomena. Vehicle instability can occur due to unexpected road profile changes, during full braking, obstacle avoidance or severe manoeuvring. Three main instability phenomena can be distinguished: the yaw-rate instability, the rollover and the jack-knife phenomenon. The main goal of this study is to develop a yaw-rate and rollover stability controller of an Autonomous Scaled Ground Vehicle (ASGV) using Nonlinear Model Predictive Control (NMPC). Open Source Software (OSS) known as Automatic Control and Dynamic Optimisation (ACADO) is used to design and simulate the NMPC controller based on an eight Degree of Freedom (8 DOF) nonlinear vehicle model with Pacejka tire model. Vehicle stability limit were determined using load transfer ratio (LTR). Double lane change (DLC) steering manoeuvres were used to calculate the LTR. The simulation results show that the designed NMPC controller is able to track a given trajectory while preventing the vehicle from rolling over and spinning out by respecting given constraints. A maximum trajectory tracking error of 0.1 meters (on average) is reported. To test robustness of the designed NMPC controller to model mismatch, four simulation scenarios are done. Simulation results show that the controller is robust to model mismatch. To test disturbance rejection capability of the controller, two simulations are performed, with pulse disturbances of 0.02 radians and 0.05 radians. Simulations results show that the controller is able to reject the 0.02 radians disturbance. The controller is not able to reject the 0.05 radians disturbance.

Acknowledgement

Thanks to God for creating nature and allowing us to freely learn from it with the aim of improving our daily lives. Thanks to my supervisor Dr. M.S. Tsoeu for his guidance, comments, advices and logical questions which opened my “eyes” throughout the study. Thanks to Mr. Dominic De Maar for entrusting me with lab equipment for the study. Thanks to University of Malawi (The Polytechnic) for financial support throughout the study. Finally, thank you to my family and friends for your support and encouragement, you are the reason I fight hard.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This first chapter is intended to provide a brief background of what this research is about. Mainly, it provides a statement of a problem to be addressed in this research. Furthermore, related works, purpose of study, objectives of study, scope and limitations, research contributions and plan of action are discussed. The chapter is closed by providing a report outline which summarizes what has been covered in the chapters that follow.

1.1 Background

With an increase of vehicles on the road, vehicle safety is becoming more important each day. Considerable work has been conducted in the past to decrease the number of vehicle accidents. One of the causes of vehicle accidents is vehicle instability phenomena. A vehicle becomes unstable when the driver is unable to control the vehicle by steering or by a throttle/braking input. Typical examples of vehicle instability are skidding (also called yaw-rate instability) and rollover of the vehicle. Between these two phenomena, rollover is the most dangerous and fatal incident.

According to the National Highway Traffic Safety Administration (NHTSA) 2014 [1], rollover occurrence accounted for over 35% of all fatalities in passenger vehicles and light trucks in 2009 in United States of America. This percentage has steadily risen over 7.5% since 1982. In 2014 rollover occurrence accounted for 32.5%, which is still very high. This high percentage is the cause for continuous research in rollover prevention schemes.

Rollover prevention is a fundamental and significant issue for vehicle dynamics and has been a topic of considerable research for a long time [2], [3], [4], [5], [6]. There are two distinct types of vehicle rollover: tripped and un-tripped rollover [7]. Tripped rollover commonly occurs when a vehicle skids and digs its tires into soft ground or hits a tripping mechanism such as a curb or guardrail with a sufficiently large lateral velocity. Manoeuvre-induced un-tripped rollover can occur during typical driving situations and poses a real threat for the vehicles with an elevated Center of Gravity (CoG). Examples are excessive speed during cornering, obstacle avoidance and severe lane change manoeuvres, where rollover occurs as a direct result of the lateral wheel forces induced during these manoeuvres [7].

Furthermore, vehicle rollover can also occur during external disturbances like side-wind. Thus, passenger vehicles with a high CoG such as light trucks (vans, pickups, and SUVs (Sport

Utility Vehicles)) are more prone to rollover accidents. Moreover, the heavy commercial vehicles with narrow track width are often involved in rollover accidents. In heavy vehicles, extra attention is drawn to vehicles carrying liquid cargo, as they are especially prone to rollover initiated by slosh forces. If the tank of such a vehicle is only partially full, slosh forces can reduce the lateral acceleration at which rollover can occur to as low as 0.3g - 0.4g [8]. During transient and steady state manoeuvres, movement of the liquid cargo results in the lateral and vertical shift of the center of gravity. Also, rollover prevention is of high importance for military vehicles, which operate in severe operational environments and manoeuvres.

A detailed classification of rollover was presented by Parenteau et al. [9]. In this paper eight different types of rollover are distinguished as follows:

- Trip-over – vehicle rolls over when during lateral motion it hits a curb, pothole etc.
- Fall-over – when vehicle rollover is caused by the lateral slope of the surface e.g. vehicle leaving the road and falling down a mountain.
- Flip-over – rollover induced by ramp or ramp-shaped down-turned object e.g. vehicle falling into a ditch.
- Turn-over – vehicle rollover initiated by lateral friction forces (this is referred to as driver-induced rollover by Solmaz et al. [10].
- Bounce-over – vehicle rollover caused by the rebound after driving onto an object.
- Climb-over – vehicle climbs over the object e.g. barrier, lifts off the ground and rolls on the side of an object opposite to the side from which it was approached.
- End-over-end – vehicle rolls about its lateral axis.
- Collision with other vehicle – collision impact causes vehicle to rollover.

Typically, a driver does not have any indication before a rollover happens and many rollover situations cannot be prevented by driver actions alone, even when they are correctly warned. Additional assistance from active anti-rollover control can mitigate the deficiency in human capability. Hence, rollover prevention systems are classified into two stages: detection of the possibility of a rollover, and development of a mitigation control algorithm. Thus, the research on rollover prevention systems has mainly focused on two areas: rollover detection systems and anti-rollover control systems.

The development of vehicle active safety systems dates back to the eighties. Examples include the Anti-lock Braking System (ABS) [11], the Traction Control (TC) system [12], and the Electronic Stability Program (ESP) [13], also known under different acronyms such as VSC for Vehicle Stability Control and IVD for Interactive Vehicle Dynamics. These “standard active safety systems” aim at assisting and improving the driver’s control over the vehicle by avoiding undesired situations, such as wheel locking in braking (ABS), tire slipping (TC), and lose of steering control (ESP). While these systems’ effectiveness in general has been widely acknowledged, they can offer little help when the driver is inattentive, which unfortunately contributes to 22 − 50% of all crashes according to some studies [14].

More recent developments in “advanced active safety systems” introduce additional sensors such as onboard cameras, radars, infrared sensors etc., and additional actuators such as active steering or active suspensions. Vehicles equipped with these systems are able to identify obstacles on the road such as pedestrians as well as the lane markers. They can perform emergency manoeuvres (mostly emergency braking) to avoid collisions or apply assist steering to avoid lane departures.

An example of vehicle with advanced active safety systems is the Volvo S60 equipped with cameras and radars in order to implement a fully autonomous braking [15]. An even more ambitious approach towards driving safety is the development of autonomous driving systems. These systems aim at driving the vehicle fully autonomous by controlling the steering, braking and throttling, examples include the Darpa Grant Challenge Car [16] and the Google Car [17].

1.2 Related Works

Rollover detection systems and anti-rollover control systems have been presented by different authors in literature. Different authors approach the rollover prevention problem from different angles. Mostly the differences are based on both detection and control of vehicle rollover. This section is intended to provide a brief summary of related works to roll prevention problem. Detailed analysis of the related works is provided in chapter 3 of this report.

In [18] a model-based roll estimator was designed to estimate the roll angle and roll rate of the vehicle. A rollover index (RI) which indicates an impending rollover was developed by a roll dynamics phase plane analysis. The rollover index was calculated using roll angle, roll rate, lateral acceleration and time to wheel lift (TTWL). A rollover mitigation control (RMC) scheme threshold was determined from the rollover index. Differential braking control law was applied to the RMC and the threshold of RMC was designed by RI.

A similar approach as in [18] is found in [19], where a unified chassis control (UCC) strategy to prevent vehicle rollover and improve vehicle lateral stability is described. Specifically, RI/lateral stability-based rollover mitigation (ROM) controller was designed to reduce the danger of rollover without loss of vehicle lateral stability. This was done by integrating electronic stability control (ESC) and continuous damping control (CDC).

In [20], [7], [21] another measure of performance, apart from rollover index, for rollover prevention is introduced. This measure is called Load Transfer Ratio (LTR). Similarly to the rollover prevention mechanisms mentioned; differential braking- based rollover controllers have been designed to keep the value of this quantity below a certain level. Specifically in [7], controllers which yield robustness to variations in vehicle speed were obtained. It was

noted that the load transfer ratio is related to tire lift-off and it can be considered as an early indicator of impending vehicle rollover.

Many of the rollover detection and prevention schemes mentioned so far act when vehicle instability is imminent, which results in aggressive actions [22]. In addition to aggressive actions, they need the vehicle to be equipped with actuators with fast speed of response which becomes impossible or extremely expensive sometimes. A common framework in the literature which can solve these problems is model predictive control (MPC).

MPC theory has been applied to a variety of process control applications [23]. Due to high computational power needed, only slow plants used to benefit from MPC. However, everyday progress in processors speed and new off-line techniques [24] together with the ability of input, states, and output constraints handling make MPC much more applicable to ordinary and fast plants. MPC is the only control technology that can systematically take into account the future predictions and safe system operating limits (constraints) in the design stage [25]. This makes it a suitable choice for systems where the system faces dynamically changing environment and has to satisfy crucial safety constraints (such as obstacle avoidance) as well as actuator constraints.

Prevention of vehicle rollover using model predictive control (MPC) is reported in literature as well. Examples of rollover prevention using MPC are found in many publications including [26], [27], [28], [29] and [22].

In [26], a linear model predictive control approach to roll stability of a scaled crash avoidance vehicle was presented. A roll stability controller (RSC) based on an eight degree of freedom dynamic vehicle model was designed. The controller was designed for and tested on a scaled vehicle performing obstacle avoidance manoeuvres on a populated test track. A rapidly-exploring random tree (RRT) algorithm was used for the vehicle to execute a trajectory around an obstacle. The vehicle examined the geographic, non-homonymic, and dynamic constraints to manoeuvre around the obstacle.

In [27], linear model predictive control (MPC) was used within a multilevel control framework; the lower level controller (LLC) consisted of two subsections: predictive steering controller and brake controller. Lower level model predictive steering controller predicted the vehicle condition during next time horizon. It intervened in driver's command to avoid an imminent rollover. Losing the driver's desired path was the penalty however. MPC of steering confined the vehicle rollover risk within the predefined limit. Rollover prevention was done online by the lower level controller. Proportional Derivative (PD) brake controller of LLC slowed down the vehicle to the desired speed of the upper level controller (ULC).

In [28], linear model predictive control (MPC) scheme was proposed that potentially could achieve both yaw stability and roll stability. The objective of the proposed MPC algorithm was to track a reference yaw rate and body side slip angle, which served the purpose of yaw

stability control. Moreover, a rollover prevention constraint was imposed to achieve roll stability. The MPC optimized the vehicle dynamics by stabilizing the yaw motion while observing the rollover prevention constraint (limited lateral acceleration), hence the yaw and roll stability could be achieved simultaneously.

In [29], a control method of switching the linear MPC controllers which uses differential braking with dead zone and active rear steer (ARS) was proposed. Simultaneously, the trade-off between rollover prevention and path tracking was highlighted through simulation results. The effectiveness of using switching controllers designed for the trade-off solution was also confirmed through MATLAB/Simulink simulation results.

In [22] a control algorithm to follow a curved road while simultaneously preventing rollover was introduced. Model predictive control was applied to minimize roll motion throughout cornering. The prediction of vehicle state was based on a four-wheel nonlinear vehicle model with roll dynamics and a brush tire model. Full braking was utilized as a control actuator to achieve an optimal balance in the trade-off between the vehicle-speed and roll motion.

1.3 Research Problem

Rollover motion of vehicles, both autonomous and manned, can occur at any point (not during cornering only) of trajectory followed by them. This motion can be due to external disturbances like wind gusts, potholes, non-flat road, during obstacle avoidance and many more. The analyses presented in [22] and [26] focus on rollover prevention only during cornering and obstacle avoidance respectively, as described in related works section.

Controlling rollover by controlling steering angle only as described in [27] and [28] mostly results in trajectory tracking error, as highlighted in their results. Using switching MPC controllers to track the trajectory as in [29] only tracks the trajectory after the manoeuvre is complete. Many publications use differential braking to prevent rollover, for example the authors in [29] prevented rollover during obstacle avoidance by using differential breaking. Differential breaking precludes yaw motion [22] hence results in large trajectory tracking error. In general, vehicles deviate from expected trajectory when prevented from rollover by means of yaw moment insertion (through steering or differential breaking). Large trajectory errors can result into causing other accidents as well, for example bumping into other vehicles.

Considering that rollover can occur at any point, the work in this study extends the idea in [22] by applying MPC to the whole path of travel of an autonomous vehicle. This full path MPC application also eliminates the task of ensuring bumpless transfer which is crucial when two or more controllers are used. In addition to constraining roll angle; yaw angle, yaw rate and inputs are constrained as well. Full braking aimed to achieve the yaw rate

required (reduce trajectory tracking error) to follow a curved path while simultaneously preventing rollover is used.

1.4 Purpose of Study

As a way of contributing to the ongoing research on autonomous vehicle safety, the purpose of this study is therefore to develop a Nonlinear Model Predictive Control (NMPC) controller capable of constraining the yaw-rate, side-slip and roll angles to maintain autonomous vehicle dynamic safety. This work aims to allow an autonomous vehicle to operate safely at a maximum possible (optimal) speed while tracking a desired path. Once these goals are met it ensures that the control system, after integrating with other sensors, can be capable of not only sensing a possible emergency situation, but also reacting to it in order to maintain autonomous vehicle safety.

1.5 Objectives of Study

The objectives of this study are as follows:

- To develop an NMPC controller which is capable of maintaining autonomous vehicle stability by constraining yaw-rate, side-slip angle and roll angle while following a defined path at an optimal speed.
- To do closed loop simulations of the controller with a nonlinear vehicle model which is capable of capturing all the dynamics of interest (yaw-rate, side-slip and roll angles).
- To analyse the developed controller's sensitivity to parameter changes (robustness) and disturbances.

1.6 Scope and Limitations

A fully autonomous vehicle would require the use of a number of different control systems in order to carry out different tasks. These systems include, but are not limited to the following: vision systems for obstacle detection and avoidance, vehicle steering, friction monitoring, skid, active suspensions, braking and cornering control systems, path generation and localization and mapping algorithms.

The scope of this research, however, is limited to designing and implementing a controller that makes an autonomous vehicle follow a specific path at optimal safe speed to avoid yaw-rate and rollover instability phenomenon. The only inputs that are used are front steering angle and all-wheel braking/throttle control. In addition to that, only robustness and disturbance rejection of the controller are analysed. Robustness check is limited to 5% difference in all parameters and 10% difference in sprung mass. For disturbance rejection,

investigation is limited to 0.02 radians and 0.05 radians pulse disturbances in roll angle (which is directly linked to rollover).

Furthermore, developing autonomous cars and testing them are very cost-intensive activities. In an attempt to have an experimental testing environment that serves as a validation but is nevertheless feasible on a reduced budget, a small-scale setup is instructive in the comparison of different algorithms for autonomous driving. Therefore, in this research the vehicle used is a scaled version which has the dynamics that resemble those of a full scale passenger vehicle as closely as possible, with the exception to its drive system. The used test bed is a 1/10th scaled Radio Controlled (RC) car driven by a brushless DC motor.

1.7 Research Contributions

There are many vehicle safety control methods available in literature as highlighted in the background and related works sections. The methods are designed in different ways based on their specific task to be accomplished. This work focuses on the design of a Nonlinear Model Predictive Control (NMPC) controller based on a 14 state nonlinear vehicle model with two inputs. To ensure vehicle stability, the NMPC controller controls the yaw-rate, side-slip angle and roll angle of an autonomous vehicle while the vehicle is following a defined reference trajectory. The following are the main contributions of this study:

- An NMPC controller for autonomous vehicles to ensure vehicle stability by constraining yaw-rate, side-slip angle and roll angle to be within the safe region during the whole path of travel. The controller is designed and simulated using an Open Source Software (OSS) known as Automatic Control and Dynamic Optimisation (ACADO) software.
- Implementation of the NMPC controller within the ACADO environment (other than generating the code), which makes it very easy to do some changes when need arises.

1.8 Plan of Action

To achieve the goal of this research, a series of tasks will be carried out. First and foremost, literature review of relevant topics and related works to this study will be done. After literature review, the modelling of the scaled platform vehicle will follow, which will then lead to system identification of the platform vehicle parameters to be used in simulations. Then the controller will be developed to control the vehicle model based on stability limits which will be identified through simulations of the vehicle model. After that, the developed controller simulations with the scaled vehicle platform model will be performed and results discussed accordingly.

To reduce project cost, free open source software (OSS) will be used in this research. Specifically, Scilab will be used during system identification, stability analysis and generating an offline trajectory. To solve the nonlinear problem of the vehicle during simulation, an OSS known Automatic Control and Dynamic Optimization (ACADO) will be used. ACADO compiled within Microsoft visual studio (MVS) community environment is used during simulations. More details on the tools and methods used are provided within their specific applicable sections in this report.

Table 1.1 is a Gantt chart showing a summary of the plan of main tasks and time allocated to them.

Table 1.1: Project Gantt chart

		YEAR ONE (2017)											YEAR TWO (2018)							
ID	ACTIVITY	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG
1	Preliminary literature review																			
2	Project proposal writing																			
3	Literature review																			
4	Dynamic vehicle modelling																			
5	Platform vehicle model identification																			
6	Control law development																			
7	Platform vehicle model contoller simulation																			
8	Dissertation writing																			
9	Dissertation submission																			

1.9 Report Outline

This report is organized as follows:

- **Chapter 1** provides a general background of the whole dissertation. In this chapter, background to study, related works, research problem, purpose, objectives, scope and limitations of the study, research contributions and plan of action are provided.
- **Chapter 2** presents literature review of the whole dissertation. The chapter starts by looking at different vehicle dynamics and tire models, vehicle stability criteria and system identification methods and algorithms. Then the chapter covers information on MPC which includes MPC solution method, stability and robustness. The chapter concludes by looking at ACADO software. Information regarding other nonlinear optimization software other than ACADO is also given including the motivation for ACADO software development.
- **Chapter 3** presents a detailed review of the related works to the study and expands on the research problem as well. The chapter starts by providing more details of the related works provided in the introduction chapter as well as commenting on the results of the related works. Then expansion of the research problem follows which concludes the chapter.
- **Chapter 4** shows a derivation of a vehicle model used in this study. The chapter begins by providing a general overview of vehicle dynamics models available, and then proceeds to motivate for a just complex enough model derived. Then the model derivation follows, which is just complex enough and include all the interested phenomena of study.
- **Chapter 5** presents system identification for the test bed platform used in this research; it covers information on experimental apparatus, experimental design and results and nonlinear least squares method within Scilab environment. The chapter concludes by validating the model.
- **Chapter 6** presents stability analysis of the platform vehicle model used in this study. In this chapter, roll angle stability limit and yaw rate stability limit are identified.
- **Chapter 7** presents the design of the NMPC controller used in this study. The chapter covers details on how the NMPC controller has been designed within ACADO environment. In addition to that, advantages and disadvantages regarding the proposed controller structure are given; mainly as compared to ACADO code generation tool.
- **Chapter 8** presents the simulation of the designed NMPC controller for the platform vehicle model. The controller robustness to parameter changes (plant model mismatch) and disturbance rejection are presented in this chapter as well. In addition to that, effects of the NMPC tuning parameters on controller performance are investigated.
- **Chapter 9** provides a conclusion and future work of the work reported in this document.

Chapter 2

Literature Review

This chapter covers literature review of relevant topics to this work. The chapter looks at different vehicle dynamics and tire models, vehicle stability criteria, vehicle system identification methods and algorithms, and finally concludes by looking at Nonlinear Model Predictive Control (NMPC) theory in conjunction with Automatic Control and Dynamic Optimization (ACADO) software.

2.1 Vehicle Dynamics Models

2.1.1 Introduction

In this section, modelling of vehicle dynamics for control purposes is addressed. For vehicle dynamics and control studies, lumped parameter models are usually employed and typically these models focus on either ride qualities or handling qualities of the vehicles. A ground vehicle with a suspension consists of two main parts: the sprung mass and the unsprung mass. The sprung mass is the mass of the body and other components supported by the suspension and the unsprung mass is the mass of suspension, wheels or tracks and other components directly connected to them, rather than supported by the suspension. The motion of a vehicle with the constraints of the road has Six Degrees Of Freedom (6 DOF), classified as follows:

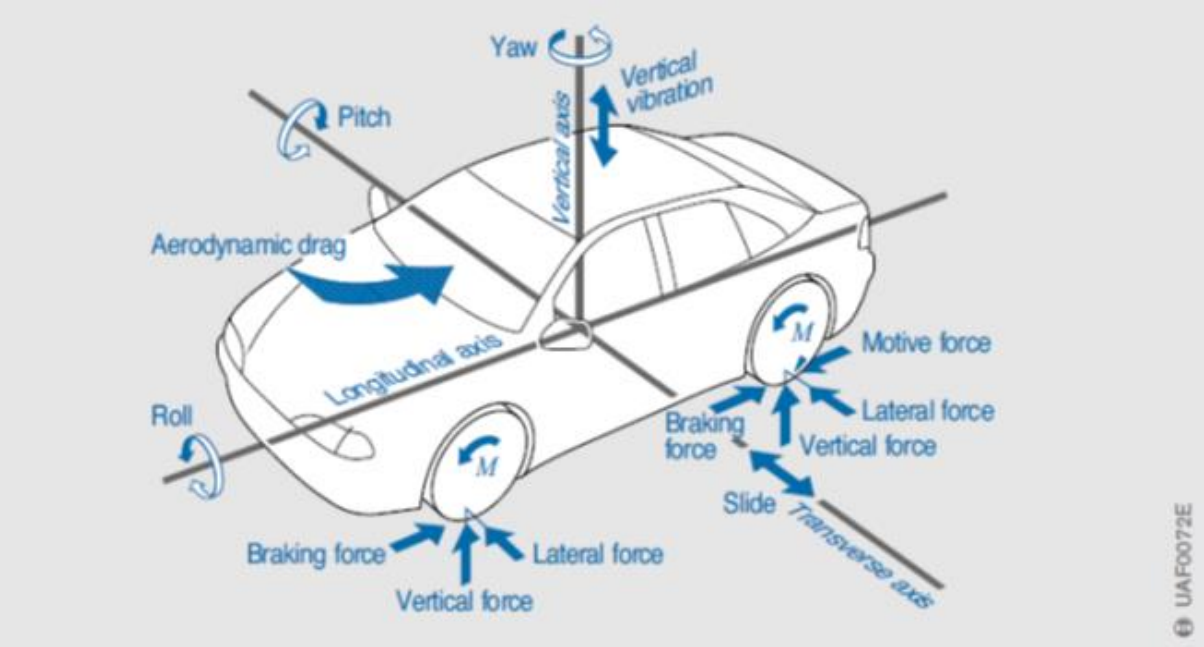

Figure 2.1: Showing vehicle main degrees of freedom [30].

- Longitudinal translation (forward and backward motion).
- Lateral translation (side slip motion).
- Vertical translation (bounce or heave motion).
- Rotation about the longitudinal axis (roll).
- Rotation about the transverse axis (pitch).
- Rotation about the vertical axis (yaw).

These DOF are shown diagrammatically in figure 2.1.

Vehicle ride is essentially related to the vehicle vertical dynamics (bounce, pitch, and roll) whereas handling is concerned with lateral dynamics (side slip, yaw, and roll). Note that roll is coupled both in ride and handling.

In order to develop the most complete model of vehicle roll behaviour and examine roll response associated with specific manoeuvring conditions, it is necessary to consider the vehicle model combining motions both in the yaw and roll planes. That is because lateral, yaw and roll dynamics are all coupled. Therefore, handling model which includes lateral, yaw, and roll dynamics is considered as a rollover model. Pitching and bouncing motion have relatively small effects on vehicle lateral dynamics, thus they are neglected in most rollover models. If pitching and bouncing are too much, they result in loss of traction, which in turn makes the models invalid for rollover studies.

Vehicle modelling is divided into two areas: tire modelling and chassis modelling [31]. Tire modelling deals with understanding the forces that arise at the tire-road contact point. Chassis modelling involves determining the behaviour of the vehicle when subjected to these external forces. This section begins with a brief introduction to tire modelling, in which some models are described. Chassis modelling is then performed yielding a number of models of different complexities.

2.1.2 Tire Models

Pneumatic Tires

Tires are the only contact point between the vehicle and the outside environment through which the vehicle can be controlled. Therefore it is of critical importance to have a clear understanding of their operation and the forces they generate. Tire characteristics are of crucial importance for the dynamic behaviour of the road vehicle [31]. Modern automobile tires are extremely complicated products to say the least; they are a composite of various rubber compounds, steel and often Kevlar and the design have been refined through decades of development [32]. Tread patterns may be highly specialized for removing water from between the tire and road or for gripping on snow and ice. The rubber itself may be sticky for maximum grip or hard for longevity. Furthermore, the inflation pressure and wear on a tire also affect the manner in which it generates force.

Due to the highly non-linear nature of the tire, modelling its behaviour is quite difficult. Despite this, many models exist, such as the Brush model [33], the LuGre model [34], and the Dugoff model [35]. Although the method by which a tire can be modelled follows a continuum, there are typically three main categories in their mathematical treatment: finite element modelling (FEM), analytical modelling, and empirical modelling.

Tire designers use FEM [36], [37] models in order to better understand how the tire behaves prior to the production of the tire itself. An accurate FEM model is both very time consuming to produce and resource intensive to complete. It also requires the use of accurate material models. Since the tire is a composite structure featuring non-linear materials such as rubber, material modelling alone is quite challenging. Although these models can accurately define tire behaviour, they are computationally intensive and are usually not the best option for real-time vehicle simulations [31].

Empirical tire models use testing equipment in order to map the tire response under many different configurations. The data is then used to develop curve fits and interpolation is used to estimate tire behaviour outside of the tested conditions. One of the most well-known empirical tire models is the Pacejka [31], or 'magic tire model'. This is widely used because of its ability to accurately define a tire response over its linear and nonlinear regions, without requiring intensive computation. While empirical models give excellent results, as far as accuracy, the cost of fitting a specific tire to a model, such as the Pacejka model, is quite significant.

Analytical tire models attempt to predict tire behaviour through the analysis of a simpler representative dynamic system. Though this is likely the least utilized method of modelling a tire, it has some advantages over the previously discussed approaches. First, one analytical model should be capable of describing the behaviour of any tire, while FEM and empirical models only give information about the specific construction being analysed. Secondly, parameterization of the tire theoretically should not require a full scale empirical test of the direct dynamic response of the tire. An example of such a model is the 'brush' tire model described in [33], [38]. This model defines the tire as a group of elastic/spring-like cylinders positioned radially outside of a circular belt. The resulting model is analytical in nature.

Based on the details in the previous paragraphs, empirical and analytical models are considered in this study (FEM models are computationally intensive for vehicle simulations). To further contrast the difference between empirical and analytical modelling, an example of each will be given. First, the empirically based Pacejka tire model will be introduced. Next an analytical model known as brush tire model is introduced.

This section begins by defining some of the terminology used when dealing with tire dynamics before presenting the two tire models. The definitions here follow those set out by the book: Tyre and Vehicle Dynamics [31] written by Hans B. Pacejka.

Tire Quantities

i. Pure Longitudinal Slip

The upright wheel rolling freely, that is without applying a driving torque, over a flat level surface along a straight line (no side slip) may be defined as the starting situation with all slip equal to zero. When a driving torque is applied, however, the condition of zero slip is no longer fulfilled and a build-up of additional tire deformation and possibly partial sliding at the contact patch may occur. As a result horizontal forces are developed. This slip quantity will serve as an input into the tire system with the resulting output being the forces developed.

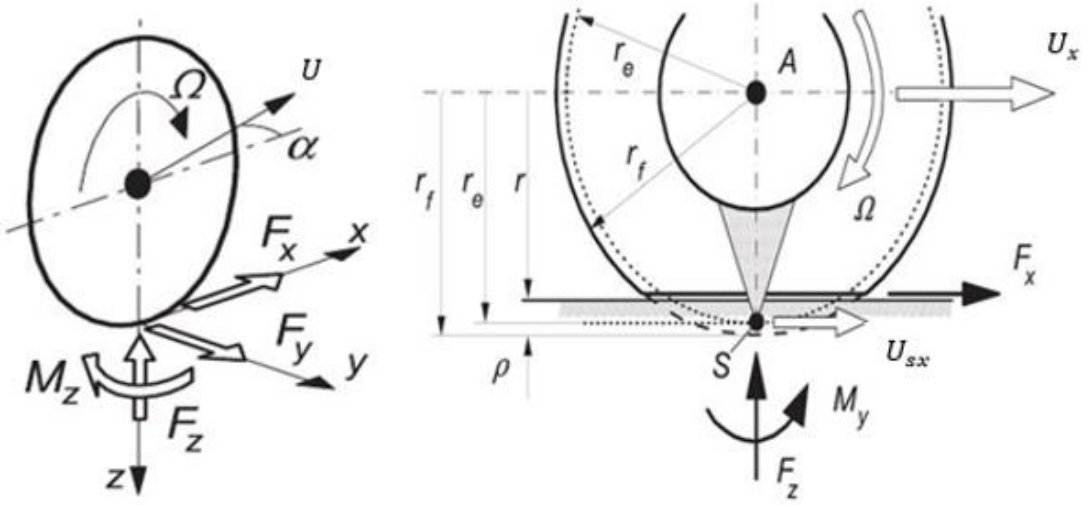

Figure 2.2: Slip angle, Forces and Moment in positive directions (left side); Effective rolling radius and longitudinal slip (right side) [31].

Figure 2.2 shows a freely rolling wheel, with forward speed U_x (the longitudinal component of the total velocity vector U of the wheel centre) and angular speed of rotation Ω can be taken from measurements. Effective rolling radius r_e is defined by dividing these two quantities:

$$r_e = \frac{U_x}{\Omega}. \tag{2.1}$$

When a torque is then applied to the wheel spin axis, a longitudinal slip arises and is defined as follows:

$$k = -\frac{U_x - r_e\Omega}{U_x}. \tag{2.2}$$

The sign of k is taken such that for a positive slip value, a positive longitudinal force (forward driving force) F_x arises. It can be noted that at wheel lock up (i.e. when $\Omega = 0$) $k = -1$, but for very large values of Ω in comparison to U_x; k can become very large. Therefore, in order to limit the value of k, the following definition is used:

$$k = \begin{cases} \dfrac{U_x - r_e\Omega}{U_x} & \text{if } U_x \geq r_e\Omega, \\ \dfrac{r_e\Omega - U_x}{r_e\Omega} & \text{if } U_x < r_e\Omega. \end{cases} \tag{2.3}$$

ii. Pure Lateral Slip

Lateral slip, similarly to the longitudinal slip, is defined by the ratio of lateral and forward components of the velocity vector U. This corresponds to the tangent of the slip angle [31]. Again the sign convention results in a positive lateral force generated by a positive (right hand axis sign convention) slip angle.

$$\tan\alpha = \frac{U_y}{U_x}. \tag{2.4}$$

The lateral slip, α, combined with the longitudinal slip, k, and vertical load, F_z, (which may be considered a given quantity that results from the normal deflection of the tire) are needed as main inputs to tire models. The output forces from the tire models are defined as follows:

$$F_x = F_x(k, \alpha, F_z), \tag{2.5}$$

$$F_y = F_y(k, \alpha, F_z), \tag{2.6}$$

$$M_z = M_z(k, \alpha, F_z). \tag{2.7}$$

Where F_x, F_y and M_z are longitudinal force, lateral force and restoring moment respectively.

iii. Combined Slip

The notion of combined slip comes about due to the limited static frictional force available to the tire at any given time. In general, the force generation on a tire is the vector sum of the lateral and longitudinal components, and must be less than or equal to the maximum available force to keep the vehicle stable. The total force is given by the following relationship in equation 2.8. The coefficient of friction between the tire and the road is μ.

$$F_{total} = \sqrt{F_x^2 + F_y^2} \leq \mu F_z \tag{2.8}$$

The relationship results in what is called a slip circle (force limit circle). This slip circle represents how force is distributed between lateral and longitudinal force. Inside this slip circle, the tire force does not reach its limit and increasing slip increases tire force. However, outside of the slip circle, increasing slip no longer generates additional force. Figure 2.3 shows this circle.

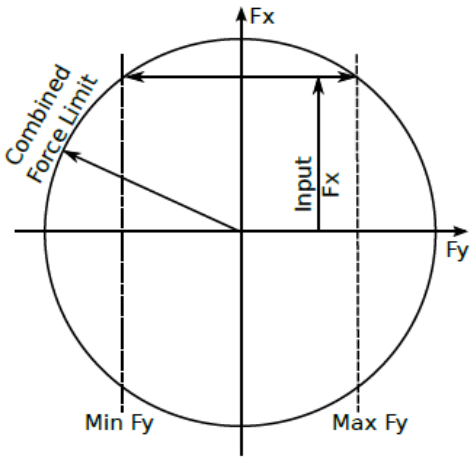

Figure 2.3: Force limit circle (slip circle) [31].

Tire Modelling

i. Pacejka Magic Formula Tire Model

The Pacejka Magic Formula Tire Model, named after its creator Hans B. Pacejka [31], is a semi-empirical tire model which uses a combination of data curve-fitting as well as physical tire properties in order to approximate the forces generated by the tire. The magic formula is not only extremely accurate in describing the measured data, but also characterizes some of the typifying quantities such as slip stiffness and peak values which permit the calculation of forces and torques in conditions which deviate from those imposed during the actual measurements [31]. Since its inception in 1987 it has become the most widely used tire model and can be found in most vehicle modelling software.

The formula is termed "magic" because there is no particular physical basis for the structure of the equations chosen, yet they fit a wide variety of constructions and operating conditions. The general form of the equation is as follows:

$$y = D\sin(C\arctan\{Bx - E(Bx - \arctan Bx)\}). \tag{2.9}$$

With

$$Y(X) = y(x) + S_v, \tag{2.10}$$

$$x = X + S_H, \tag{2.11}$$

$$B = \frac{C_{F\alpha}}{CD}, \tag{2.12}$$

$$D = \mu F_z, \tag{2.13}$$

$$C_{F\alpha} = BCD. \tag{2.14}$$

Where Y is the output variable F_x, F_y or possibly M_z and X the input variable $\tan(\alpha)$ or k and B is the stiffness factor, C the shape factor, D the peak value, E the curvature factor, S_H the horizontal shift, and S_V the vertical shift. Figure 2.4 illustrates the meaning and effect of these parameters graphically.

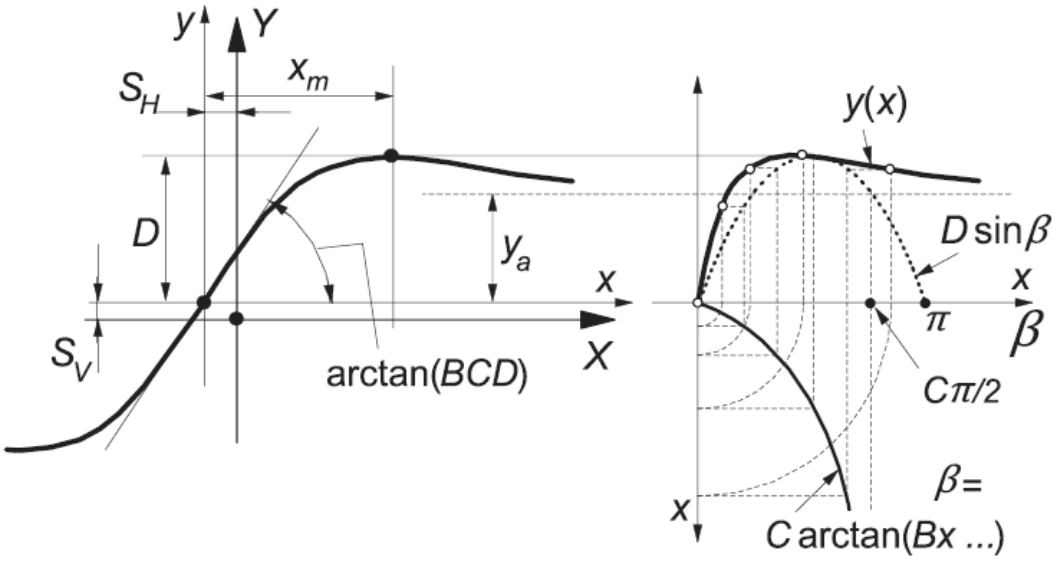

Figure 2.4: Curve produced by the magic formula [31].

ii. Brush Tire Model

The brush tire model describes the structure of a tire using a row of elastic bristles which touch the road plane and can deflect in a direction parallel to the road surface. The assumption is that the slip, both longitudinal and lateral, relies on the compliance of "bristles" or treads elements of the tire which represents the elasticity of the real tire carcass and tread. As can be seen in the figure 2.5, when the wheel speed vector U shows an angle with respect to the wheel plane, side slip occurs. When the wheel velocity of revolution Ω multiplied with the effective rolling radius r_e is not equal to the forward component of the wheel speed $U_x = U\cos\alpha$, the fore-and-aft slip occurs. Under these slip conditions the corresponding forces are developed.

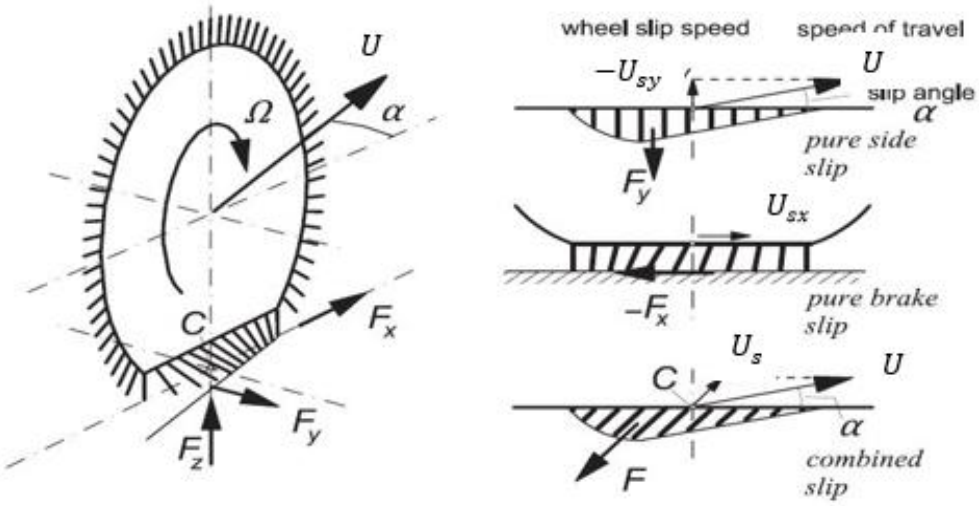

Figure 2.5: View of driven and side-slipping tire and the tire under different slip conditions [31].

As seen in figure 2.5, the tread elements move from the leading edge (right hand side of figure) to the trailing edge. The tip of each element will remain adhered to the ground as long as static friction allows, that is, it will not slide across the road surface. Simultaneously the base point, which is fixed to the tire carcass, moves backward with the linear speed of rolling ($r_e\Omega$) with respect to the wheel axis located at the contact patch centre labelled C in the diagram.

The resulting deflection of the tire bristles varies linearly with distance from the leading edge and the tips form a straight contact line which is parallel to the wheels velocity vector U . The maximum force generated by this tire model is dependent on three variables namely: The constant coefficient of friction μ, the vertical (normal) force distribution F_z and the stiffness of the tread elements $c_p y$. The pressure distribution, and consequently the maximum deflection, is assumed to have a parabolic distribution and therefore as soon as the straight contact line intersects this parabolic distribution sliding will start. This intersection can easily be seen in figure 2.6. The remaining part of the contact line will coincide with the parabola for the maximum possible deflection.

At increasing slip angle, the side force that is generated will increase. The distance of its line of action behind the contact center is termed the pneumatic trail t. As the slip increases, the deformation shape becomes more symmetric and, as a result, the trail gets smaller. This is because the point of intersection moves forward, thereby increasing the sliding range and decreasing the range of adhesion. This continues until the wheel speed vector runs parallel to the tangent to the parabola at the foremost point. Then, the point of intersection has reached the leading edge and full sliding starts to occur. The shape has now become fully symmetric and the side force attains its maximum.

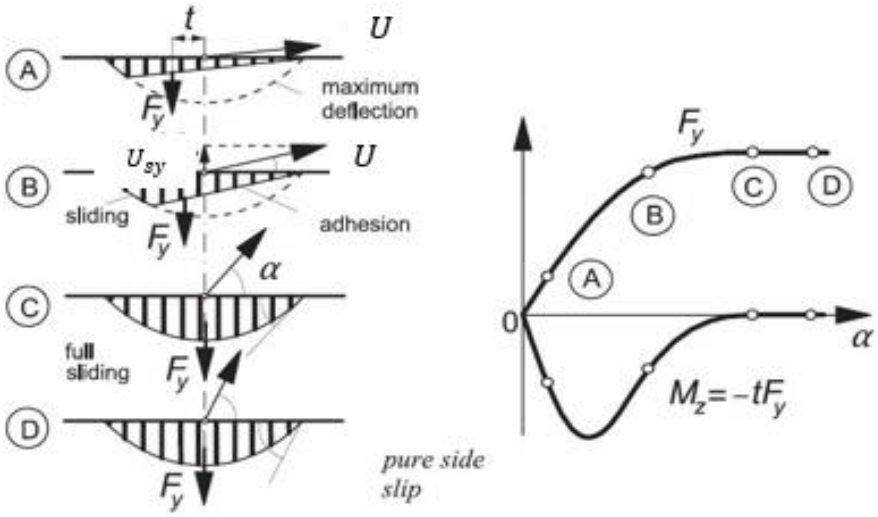

Figure 2.6: Left: the tire at pure side slip, from small to large slip angle. Right: the resulting side force and aligning torque characteristics [31].

There are different mathematical models for brush tire models developed by different users for different purposes. An example is found in [32] where the model is parameterized by the linear cornering stiffness (C_α), the peak force (μF_z), and the ratio of the sliding friction coefficient to the peak coefficient (R_μ). The relationship between the slip angle and the tire force is given by the following expression if the tire is operating at a slip angle equal to or less than the angle of full sliding:

$$F_y = -C_a \tan\alpha + \frac{C_\alpha^2}{3\mu F_z}(2 - R_\mu)\tan\alpha|\tan\alpha| - \frac{C_\alpha^3}{9(\mu F_z)^2}\left(1 - \frac{2}{3}R_\mu\right)tan^3\alpha. \quad (2.15)$$

The angle of full sliding is

$$\alpha_{slide} = \frac{3\mu F_z}{C_\alpha}. \quad (2.16)$$

This angle can be found analytically using the assumed parabolic pressure distribution and triangular force demand on the tire. Once beyond this angle, the tire force is assumed to be equal to

$$F_y = \mu_s F_z. \quad (2.17)$$

This is the maximum sliding friction force. Since the tire force equation is a cubic polynomial in tan α, once the sign of $\tan\alpha$ is known, it is possible to reverse the relationship and calculate slip angle from lateral force, but the solution is unique only if an additional assumption that the tire is operating at an angle equal to or below the slip angle corresponding to peak force is added.

2.1.3 Vehicle Chassis Models

The tire models described in the previous section are used to obtain the contact forces between the tires and the road surface. These forces constitute the inputs to the chassis model, which describes the motion of the vehicle in space. In this section, several chassis models of varying complexity will be derived. Models of different complexities are required for different applications, such as control design, reference generation and simulation. In general, the process of modelling is extremely important for gaining an understanding of the system, which is vital when choosing a control strategy. The section begins with an introduction to the coordinate systems used in the representation of the vehicle's motion. Different chassis models are presented soon after the coordinate system.

Coordinate System

In order to facilitate the derivation of the equations of motion, it is useful to define a number of coordinate systems. This allows a more systematic approach to modelling, which is particularly important when dealing with more complex models. Figure 2.7 illustrates a right-hand coordinate system. *Roll, pitch* and *yaw* are defined as rotations around the x, y and z axes respectively.

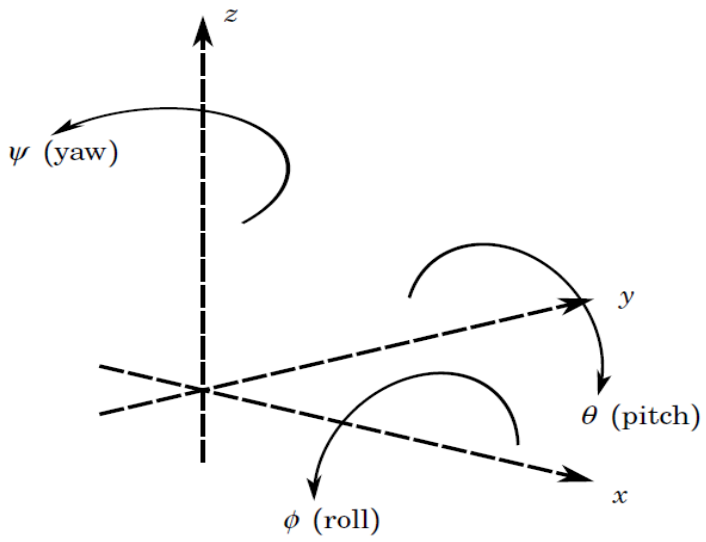

Figure 2.7: The coordinate system axes.

The x axis corresponds to the longitudinal axis of the vehicle, positive in the forward direction. The y axis corresponds to the lateral axis, positive to the left. The z axis corresponds to the vertical axis, positive upwards. Roll, pitch and yaw are defined around the x, y and z axes respectively.

Nonlinear Bicycle Model

The bicycle model shown in figure 2.8 is a simplified chassis model for a rear wheel drive four wheel vehicle. The model only focuses on planar dynamics of the vehicle; the four wheels are combined into just a single track, thereby neglecting roll dynamics. This model is

a three state model only and is used to describe the rotational (yaw) motion as well as the longitudinal and lateral velocities of the vehicle.

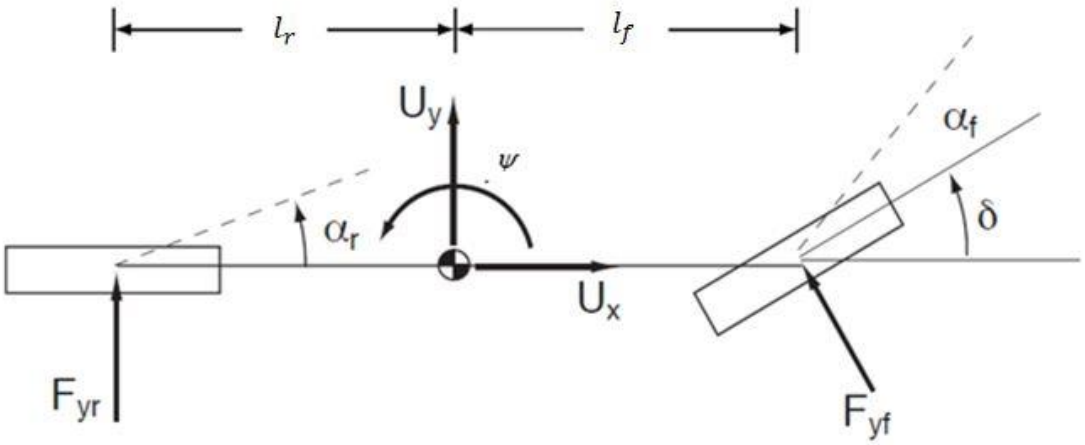

Figure 2.8: Nonlinear bicycle model [32]. Shown are rear tire lateral force (F_{yr}), front tire lateral force (F_{yf}), rear tire slip angle (α_r), front tire slip angle (α_f), steering angle (δ), longitudinal velocity (U_x), lateral velocity (U_y), yaw angle (Ψ), distance of centre of gravity from rear axle (l_r) and distance of centre of gravity from front axle (l_f).

Beyond neglecting the roll dynamics, there are a few other key assumptions made. The mass of the vehicle is considered to be located entirely in the rigid base of the vehicle chassis. Furthermore, the slip and steering angles for the left and right wheels are considered to be the same; this assumption is made based on the vehicle travelling at typical driving speeds and traversing corners of moderate radius. The forces can therefore be lumped together as a single equivalent wheel as shown in the figure 2.8. For the purposes of friction estimation and lateral force generation a constant longitudinal velocity is often used [32]. However, from previous tire modelling section it is seen that the longitudinal tire forces play a large role in how the lateral forces are generated and therefore they will be included the derivations. The equations of motion of the bicycle model are derived from first principles by writing force balance equations in the vehicle x and y - coordinate frames as well as a moment balance equation about the z - axis.

$$ma_x = F_{xf} + F_{xr}, \tag{2.18}$$

$$ma_y = F_{yf} + F_{yr}, \tag{2.19}$$

$$\ddot{\psi} I_{zz} = l_f F_{yf} - l_r F_{yr}. \tag{2.20}$$

The vehicle sideslip angle is calculated as

$$\beta = \tan^{-1}\left(\frac{U_y}{U_x}\right). \tag{2.21}$$

By taking into account the vehicle steering angle the acceleration and moment equations can be rewritten in a form which define the vehicle states as follows:

$$a_x + \dot{\psi} U_y = \frac{1}{m}\left(F_{xr} + F_{yf} \sin\delta\right), \tag{2.22}$$

$$a_y + \dot{\psi} U_x = \frac{1}{m}\left(F_{yf} \cos\delta + F_{yr}\right), \tag{2.23}$$

$$\ddot{\psi} = \frac{1}{I_{zz}}\left(l_f F_{yf} \cos\delta - l_r F_{yr}\right). \tag{2.24}$$

The parameters m , I_{zz} and δ are the mass of the vehicle, the yaw moment of inertia and the steering angle of the front wheels respectively. The forces F_{xr}, F_{yr} and F_{yf} are the forces generated by the tires at the front and rear axles and are highly nonlinear at elevated levels of lateral acceleration. Any of the tire models may be used to describe the force generation characteristics. In all the cases, the slip angles of the front and rear tires of the vehicle are required to be used as inputs to these models. These slip quantities are given as follows:

$$\alpha_f = \tan^{-1}\frac{U_y + l_f \dot{\psi}}{U_x} - \delta, \;\; \alpha_r = \tan^{-1}\frac{U_y - l_r \dot{\psi}}{U_x}, \qquad k_r = \frac{r_w \Omega - U_x}{U_x}. \tag{2.25}$$

It can be noted at this point that since the vehicle is rear wheel drive only, there is no longitudinal forces developed at the front of the vehicle. The steering input (δ) is also included in these expressions.

Linear Bicycle Model

The Linear Bicycle model described in this section lacks sufficient accuracy in the nonlinear regions of operation and therefore cannot be used for high speed vehicle control. However, when operating at low slip it can be noted that the tire force output is fairly linear with change in slip angle and therefore this model can be used to perform state estimation at low slip. The simplicity of this model allows for fast computation and gives a good understanding of driver intent as most driving occurs at low levels of slip. This model utilizes a linear coefficient for tire force generation which is called cornering stiffness coefficient. Using the equations derived for the nonlinear bicycle model the following linearization are made to come up with this linear model:

$$\sin\theta = \theta, \tag{2.26}$$

$$\cos\theta = 1, \tag{2.27}$$

$$\tan\theta = \theta, \tag{2.28}$$

$$F_y = -C_\alpha \alpha_{f,r}, \tag{2.29}$$

$$F_x = -C_k \kappa_r. \tag{2.30}$$

The linearized model presented also uses the assumption that the vehicle forward velocity remains constant i.e. $a_x = 0$. The equations of motion from first principles are therefore rewritten as follows:

$$ma_x = 0, \qquad ma_y = F_{yf} + F_{yr}, \qquad \ddot{\psi} I_{ZZ} = l_f F_{yf} - l_r F_{yr}. \tag{2.31}$$

Using the small angle approximations listed in equations 2.26 to 2.28, the vehicle side slip angle can be given as

$$\beta = \tan^{-1}\left(\frac{U_y}{U_x}\right) \quad \text{or} \quad \beta \cong \frac{U_y}{U_x}. \tag{2.32}$$

The rate of change of this sideslip is given by

$$\dot{\beta} = \frac{\dot{U}_y}{\dot{U}_x}. \tag{2.33}$$

The vehicle lateral acceleration in the inertial frame can be described as

$$a_y = \dot{U}_y + \dot{\psi} U_x. \tag{2.34}$$

Using the expressions in equations 2.32, 2.33 and 2.34, the equations presented in equation 2.31 can be rearranged to solve for the state equations as follows:

$$m\left(\dot{U}_y + \dot{\psi} U_x\right) = F_{yf} + F_{yr}, \qquad \ddot{\psi} I_{zz} = l_f F_{yf} - l_r F_{yr}, \tag{2.35}$$

$$\dot{\beta} = \frac{F_{yf} + F_{yr}}{m U_x} - \dot{\psi}, \qquad \ddot{\psi} = \frac{l_f F_{yf} - l_r F_{yr}}{I_{zz}}. \tag{2.36}$$

With parameters m and I_{zz} representing the vehicle mass and yaw inertia respectively. In order to use linear force generation, front and rear lateral slip angles are defined as follows:

$$\alpha_f = \beta + \frac{l_f \dot{\psi}}{U_x} - \delta, \qquad \alpha_r = \beta - \frac{l_r \dot{\psi}}{U_x}. \tag{2.37}$$

Using the linear force relationship shown in equations 2.29 and 2.30 the front and rear lateral forces can be expressed as follows:

$$F_{yf} = -C_{\alpha,f}\left(\beta + \frac{l_f \dot{\psi}}{U_x} - \delta\right), \qquad F_{yr} = -C_{\alpha,r}\left(\beta - \frac{l_r \dot{\psi}}{U_x}\right). \tag{2.38}$$

Where $C_{\alpha,f}$ and $C_{\alpha,r}$ are the front and rear cornering stiffness respectively.

Finally the forces shown in equations 2.38 are substituted into equation 2.36 and the state equations are rewritten in typical state-space form as $\dot{x} = Ax + B\delta$, with $x = [\beta \quad \dot{\psi}]^T$,

$$A = \begin{bmatrix} \frac{-(C_{\alpha f} + C_{\alpha r})}{mU_x} & \frac{-(l_f C_{\alpha f} - l_r C_{\alpha r})}{mU_x^2} - 1 \\ \frac{-(l_f C_{\alpha f} - l_r C_{\alpha r})}{I_{zz}} & \frac{-(l_f{}^2 C_{\alpha f} + l_r{}^2 C_{\alpha r})}{I_{zz} U_x} \end{bmatrix}, \qquad B = \begin{bmatrix} \frac{C_{\alpha f}}{mU_x} \\ \frac{l_f C_{\alpha f}}{I_{zz}} \end{bmatrix}. \tag{2.39}$$

Two-Track Model (Four Wheel Yaw-Roll Model)

Two-track vehicle model is represented in figure 2.9. The four wheels of the vehicle are connected to each other by a rigid base, while the body of the vehicle is represented by an inverted pendulum that rolls about an imaginary axis connecting the front and rear of the vehicle. Rotational states for all four wheels are included and weight transfer between the wheels is calculated. The effects of the suspensions are incorporated into the roll motion of the inverted pendulum and the weight transfer between the wheels, making this a relatively complete description of the vehicle behaviour.

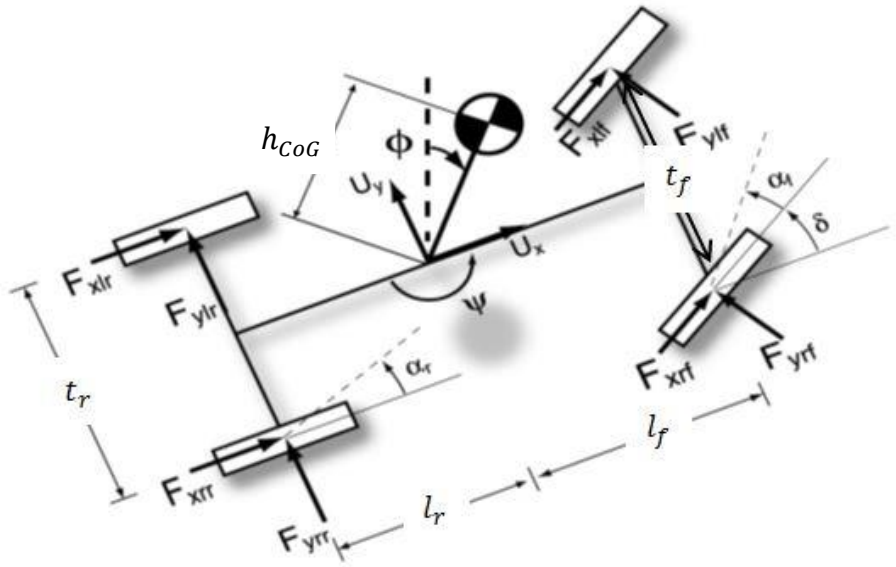

Figure 2.9: Two track model showing lateral and longitudinal tire forces, lateral and longitudinal velocities, roll and yaw angles, and slip and steering angles [32].

In the figure 2.9 the following can be seen: l_f and l_r are the distances from the front and rear axles to the centre of gravity (CoG) respectively, t_f and t_r are the front and rear vehicle track width respectively (assumed to be of same track width) and h_{CoG} is the height of the CoG. Front and rear slip angles are represented by α_f and α_r respectively. The mass of the vehicle is m; which is assumed to have no unsprung mass, and I_{zz}, I_{xx}, I_{yy} are the moment of inertia about the roll, pitch and yaw axes respectively. K_ϕ and b_ϕ are spring stiffness and damping coefficient respectively. Products of inertia are assumed to be zero in this model,

though the equations for a similar model with non-zero yaw-roll product of inertia may be found in [39].

The equations for the yaw, roll, and sideslip derivatives of the vehicle are:

$$m\dot{U}_x - m\dot{\psi}U_y + mh_{CoG}\ddot{\psi}\sin\phi + 2mh_{CoG}\dot{\psi}\dot{\phi}\cos\phi = Q_x, \tag{2.40}$$

$$m\dot{U}_y + m\dot{\psi}U_x - mh_{CoG}\ddot{\phi}\cos\phi + mh_{CoG}\left(\dot{\psi}^2 + \dot{\phi}^2\right)\sin\phi = Q_y, \tag{2.41}$$

$$\ddot{\psi} = \frac{Q_\psi - Q_x h_{CoG}\sin\phi - 2\left(I_{yy-}I_{zz}\right)\dot{\psi}\dot{\phi}\sin\phi\cos\phi}{I_{yy}sin^2\phi + I_{zz}cos^2\phi}, \tag{2.42}$$

$$\ddot{\phi} = \frac{Q_y h_{CoG}\cos\phi - K_\phi\phi - b_\phi\dot{\phi} + mgh_{CoG}\sin\phi + \left[\left(I_{yy-}I_{zz}\right)\dot{\psi}^2 - mh_{CoG}{}^2\dot{\phi}^2\right]\sin\phi\cos\phi}{I_{xx} + mh_{CoG}{}^2 sin^2\phi}, \tag{2.43}$$

$$Q_x = \left(F_{xlf} + F_{xrf}\right)\cos\delta - \left(F_{ylf} + F_{yrf}\right)\sin\delta + F_{xlr} + F_{xrr}, \tag{2.44}$$

$$Q_y = \left(F_{ylf} + F_{yrf}\right)\cos\delta - \left(F_{xlf} + F_{xrf}\right)\sin\delta + F_{ylr} + F_{yrr}, \tag{2.45}$$

$$\begin{aligned} Q_\psi = {} & l_f\left[\left(F_{ylf} + F_{yrf}\right)\cos\delta - \left(F_{xlf} + F_{xrf}\right)\sin\delta\right] - l_r\left[F_{ylr} + F_{yrr}\right] \\ & - \frac{t_f}{2}\left[\left(F_{xlf} - F_{xrf}\right)\cos\delta - \left(F_{ylf} - F_{yrf}\right)\sin\delta + F_{xlr} - F_{xrr}\right]. \end{aligned} \tag{2.46}$$

Complete derivation of the equations 2.40 to 2.46 of a two-track model is done in chapter 3 of this report. The derivation includes all other parameters that are not captured in the equations.

2.2 System Identification and Parameter Estimation Methods

2.2.1 Introduction

After vehicle systems modelling review in the previous section, the next step is to identify model parameters. In this section, system identification, parameter estimation and optimisation methods are reviewed. Some of the optimisation methods (algorithms), highlighted in this section, are implemented in ACADO and Scilab Open Source Software (OSS) which are used in this study. The work in this section mainly follows from the work done in [40] and references therein.

A *model* is defined as a mathematical relationship between a system's input and output variables. A model tries to emulate the 'essential aspects' of the system behavior, simplified by choosing the most significant properties. So, modeling techniques can be classified as follows:

- *A priori* modeling, ***white-box*** or morphological modeling; by making simple experiments to inquire into the physical or chemical laws involved.
- *A posteriori* modeling or ***black-box*** modeling; by building a model based only on data (data-driven) without having previous knowledge of the system. The model describes how the outputs depend on the inputs, not how the system actually is, and characterizes the system dynamics (delays, speed, oscillations, and others), though the physical interpretation of the results is not straightforward.
- ***Grey-box*** modeling; is an intermediate technique when peculiarities of internal laws are not entirely known, so it is based on both insight into the system and on experimental data analysis.

System identification tries to estimate a *black* or *grey* model of a dynamic system based on observing input-output from experimental data. The model has a general form which involves a number of ordinary differential or difference equations and an associated set of parameters which have to be estimated. In general, the structure (as defined by the number of differential equations and the form of any associated algebraic relationships) also involves uncertainties and the most appropriate structural form may have to be established from measured response data.

The identification problem requires a set of model structures, a validation criterion and an aim [41]. Some examples of identification aims are:

- To design control strategies for a particular system (e.g., in optimizing a vehicle operation).
- To analyze the properties of the system (e.g., quantity rates in a medication reaction).
- To forecast the evolution of the system.
- To identify hidden factors influencing a system.
- To improve the internal knowledge of the system.
- To identify the interaction between coupled systems.

2.2.2 Optimization Techniques in System Modeling

The most widely used system identification approach is based on *least squares* minimization of the difference (error) between the model response and the measured system response [40]. The process of deciding on the most appropriate structure for the model usually involves background knowledge and physical understanding of the system under investigation, as well as examination of the available response data. Once an initial model structure has been established and uncertainties in that chosen structure have been critically assessed, the parameters of the model can be adjusted in an iterative fashion using

a specified optimization cost function. The iterative processes of parameter adjustment continue until the responses of the model match those of the real system to some pre-defined level based upon values of the chosen optimization criterion.

Dynamic models used in practical engineering applications are usually nonlinear in the parameters [40]. One very important factor in nonlinear optimization is the choice of the initial parameter set. Although a random or arbitrary initial set of parameters may lead to convergence to an optimum, the selection of a favorable set of initial parameters on the basis of prior knowledge can increase the speed of convergence considerably.

Nonlinear optimization methods can be classified as *Local* and *Global* methods [42]. Although they converge to local optima, local methods often converge to points that are close to the initial parameter set, particularly with methods in which search directions are obtained from first and second-order derivative information. Such algorithms thus tend to become stuck at a local minimum or maximum and an extremum in another part of the parameter space may be neglected. Global nonlinear optimization methods can overcome this type of difficulty and rely on the inclusion of random components that help the algorithm to avoid becoming trapped at local optima. Well known global optimization techniques include *simulated annealing* (SA) described in [43] and [44], and *evolutionary* algorithms such as the *genetic algorithm* (GA) of which details can be found in [45].

Local optimization approaches that employ gradient information are widely used. The simplest gradient-based method is the *Steepest Descent* approach. This method does not require second-order derivatives of the loss function, but is known to converge slowly. *Newton's method* involves use of the inverse of the Hessian matrix and depends on second-order derivatives, which may introduce significant computational overheads. Newton's method is also computationally demanding because it involves matrix inversion. Use of the *Quasi-Newton method* reduces the computational complexity by replacing the inverse Hessian by an approximation.

The Newton and Quasi-Newton methods have good convergence properties on the basis of the number of iterations but for large problems such methods are computationally demanding. *Conjugate-Gradient* methods, such as the *Fletcher-Reeves* algorithm, provide an alternative approach to local optimization that can be less computationally demanding. Instead of using the Hessian matrix or an approximation to the Hessian, conjugate-gradient methods involve an approach where an estimate of the search direction is computed more directly. Conjugate gradient methods typically require more iterations than the Quasi-Newton and Newton methods to converge to an optimum. However, due to their computational simplicity, the overall speed of these algorithms is better.

Nonlinear least squares methods are preferred for cases in which the loss function is of the sum-of-squares type. Two well-used nonlinear least-squares methods are the *Gauss-Newton*

method and the *Levenberg-Marquardt* method as described in [46]. Gauss-Newton algorithm is closely associated with the general and modified forms of the *Newton-Raphson* algorithm for solution of numerical search problems.

The simplest general-purpose nonlinear local optimization techniques are termed *Direct Search* methods [42] and make use only of loss function values in their search for local optima. Such methods include the *Simplex Search*, *Hooke-and-Jeeves* and *Nelder-Mead* methods. These methods are typically rather slow to converge and are often only used if the derivatives of the loss function are not available or can be estimated only at considerable computational cost.

2.3 Vehicle Stability Criteria

2.3.1 Introduction

After covering system identification in the previous section, in this section, analysis of the vehicle rollover phenomenon is performed. The aim of the analysis is to give insight into the mechanisms that cause rollover, and to determine how detection of an imminent rollover event should be performed. The section begins with simple static analysis, including simple stability analysis. Methods of rollover detection are then discussed. The work presented here is similar to work in [47] and references therein.

2.3.2 Static Rollover Analysis

The underlying cause of untripped vehicle rollover accidents is the rotational motion occurring when a vehicle makes a turn [48]. Figure 2.10 illustrates a vehicle performing a turn with a radius of curvature ρ. In order to maintain the curved trajectory, a force directed towards the center of rotation must act upon the center of gravity (CoG/CG) of the vehicle. Another way of considering this is to use the method of D'Alembert, in which accelerations are represented by pseudo-forces. D'Alembert's method allows dynamics problems to be viewed as statics problems. Figure 2.11 shows the pseudo-force ma_y acting on the CoG of a vehicle performing a turn. Note that the pseudo-force acts in the opposite direction to the acceleration that it replaces, that is, it is directed radially outwards from the center of rotation. The external forces acting on the vehicle act at the road- tire contact point, not the CoG, meaning that a resulting moment acts on the vehicle. The magnitude of the resulting moment depends on the height of the CoG above the road. A higher CoG gives a larger moment. This moment is counteracted by a moment due to the reaction (normal) forces acting on the tires on the outside of the turn. This moment depends on the track width of the vehicle (the distance between inner and outer wheels). Clearly, if the moment due to the rotational motion of the vehicle exceeds the moment due to the normal forces on the tires, then the vehicle will begin to roll.

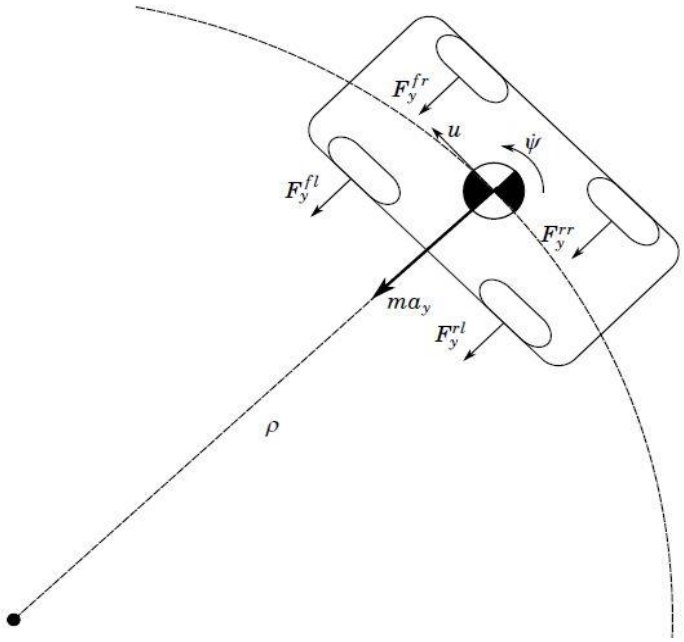

Figure 2.10: Illustration of a vehicle driving along a curved trajectory with radius of curvature ρ [47].

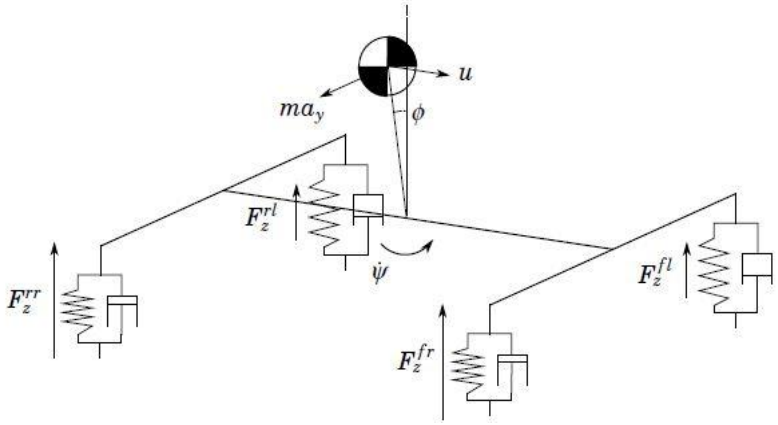

Figure 2.11: Illustration of the pseudo-force ma_y acting on the vehicle's centre of gravity [47].

A static condition for rollover can be derived from consideration of the resultant force vector acting on the CoG. If the line of action of the force lies outside the contact point of the outside wheels, then rollover occurs. Figure 2.12 illustrates the situation in the case of a vehicle without suspension. In this case, the condition for rollover to occur is:

$$ma_y h_{CoG} > mgt_r, \tag{2.47}$$

$$a_y > \frac{gt_r}{h_{CoG}}. \tag{2.48}$$

It is easy to see from equation 2.48 that the ratio of track width t_r to the height of the center of gravity h_{CoG} determines the lateral acceleration necessary for rollover to occur. It is also worthy of note that the vehicle mass m does not appear in the condition. Only the geometry of the vehicle is important. Figure 2.13 illustrates the slightly more complicated

case of a vehicle in which suspension elements are taken into account. In this case the condition for rollover to occur is given by:

$$a_y > \frac{g(t_r - h_{CoG} \sin \Phi)}{h_{CoG} \cos \Phi}. \tag{2.49}$$

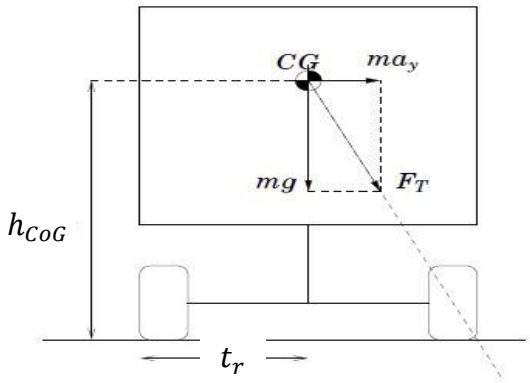

Figure 2.12: Illustration of the rollover limit for a vehicle with suspension elements neglected. The line of action of the resultant force acting on the CG passes through the contact point of the tires on the outside of the turn [47].

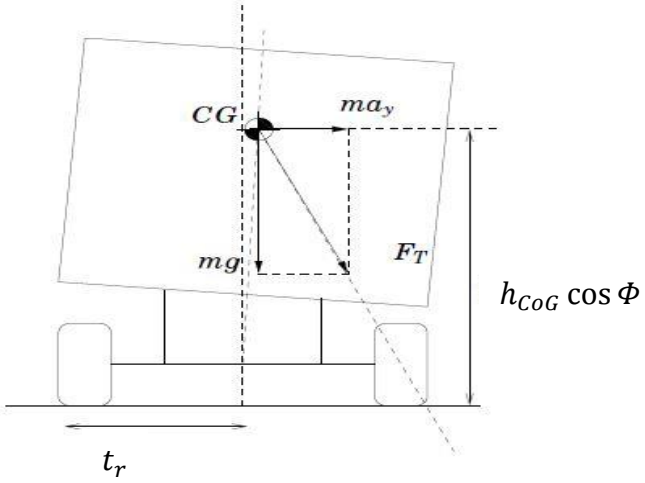

Figure 2.13: Illustration of the rollover limit for a vehicle with suspension elements taken into account. The changing position of the CG implies that the lateral acceleration required to produce rollover becomes smaller for larger roll angles [47].

2.3.3 Stability

When the lateral acceleration threshold obtained from analysis of figure 2.13 is exceeded, roll motion of the vehicle ensues. Figure 2.14 illustrates the simple case of a vehicle without suspension after the onset of rollover. To gain insight into the nature of rollover accidents, it is interesting to perform some stability analysis for this simplified model. By resolving the weight mg and the pseudo-force ma_y into components in the vehicle-fixed y and z directions, the following dynamics are obtained:

$$m(a_y t_r \sin\phi + a_y h_{CoG} \cos\phi + g h_{CoG} \sin\phi - g t_r \cos\phi) = I\ddot{\phi}. \quad (2.50)$$

It is clear from equation 2.50 that the resultant roll moment becomes larger as the roll angle ϕ increases. Once rollover has begun, the magnitude of the lateral acceleration required to sustain it decreases. The effect of the lateral acceleration is replaced by the component of the weight acting along the vehicle's y axis.

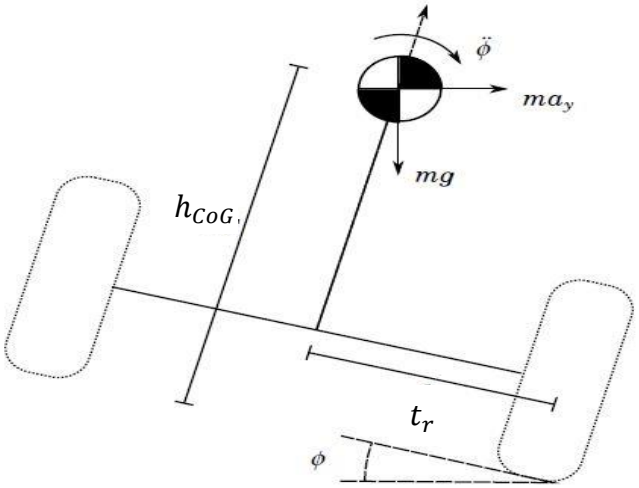

Figure 2.14: Illustration of vehicle after the onset of rollover [47].

2.3.4 Load Transfer

An important phenomenon in the study of rollover is load transfer. Load transfer refers to the shift in distribution of the vehicle's weight between the wheels [49]. This has an important effect on the forces acting on the vehicle, due to the fact that the maximum achievable friction force for each tire depends on the normal force acting on the tire.

Lateral Load Transfer

Lateral load transfer is the change in normal force acting on the tires due to both the acceleration of the CoG, and the shifting of position of the CoG in the y direction due to the movement of the suspension. Figure 2.15 illustrates lateral load transfer in the vertical plane.

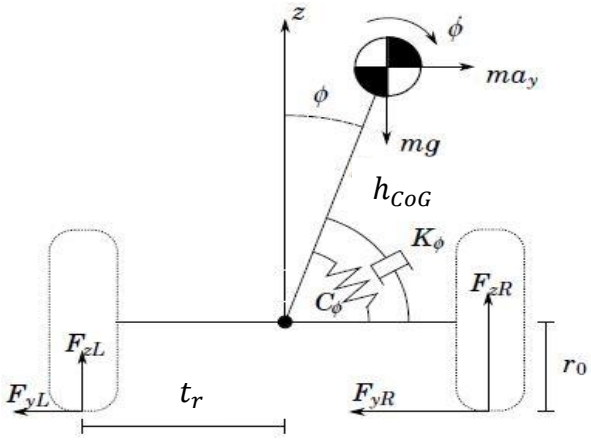

Figure 2.15: Lateral load transfer illustrated in the vertical plane [47].

Longitudinal Load Transfer

In addition to lateral load transfer, longitudinal load transfer can occur, due to acceleration or deceleration in the longitudinal direction. Longitudinal load transfer occurs between the front and rear axles of the vehicle. The total resultant load transfer for each tire is the sum of the lateral and longitudinal load transfer. Figure 2.16 illustrates longitudinal load transfer. The simultaneous effect of lateral and longitudinal load transfer is illustrated in figure 2.17.

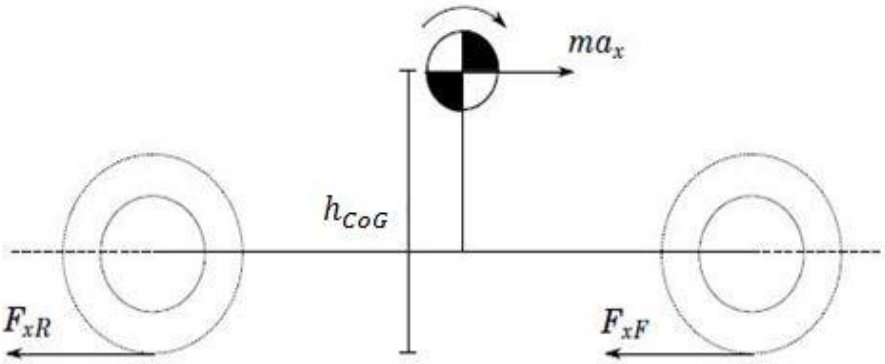

Figure 2.16: Illustration of longitudinal load transfer during braking [47].

Load Transfer Ratio

The Load Transfer Ratio (LTR) can be defined as the difference between the normal forces on the right and left hand sides of the vehicle divided by their sum, mathematically it is written as:

$$LTR = \frac{F_{ZR} - F_{ZL}}{F_{ZR} + F_{ZL}}. \tag{2.51}$$

The LTR is defined for the entire vehicle. However, it is important to note that the longitudinal load transfer has the effect of decreasing the normal forces at the rear wheels during cornering (this is true even if braking is not performed, since the lateral forces have a component in the negative x direction). This effect implies that the rear wheel on the inside of the turn will be the first to lose contact with the road during a rollover.

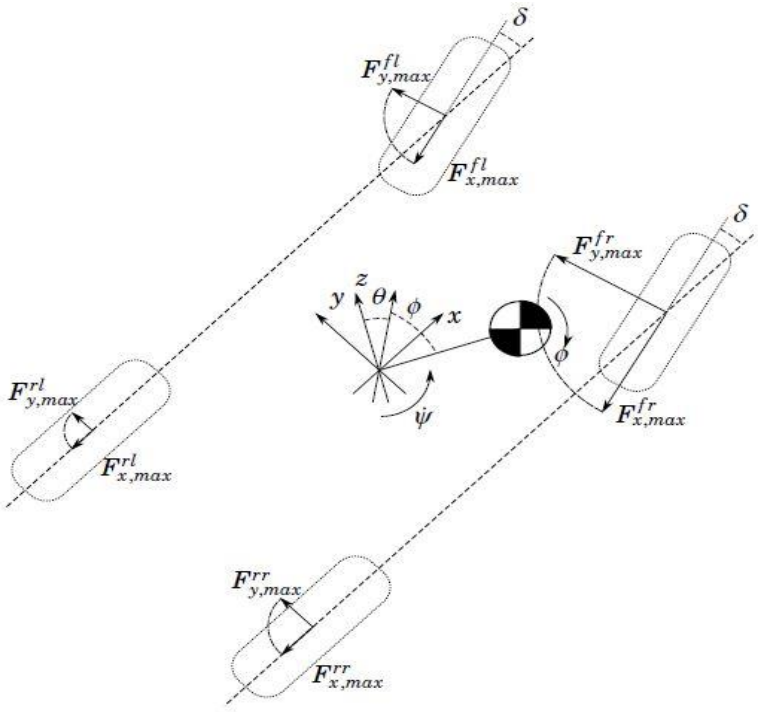

Figure 2.17: Illustration of the effects of load transfer on attainable tire forces during simultaneous cornering and braking [47].

2.3.5 Methods for Vehicle Rollover Detection

A number of methods for rollover detection have been suggested in the literature, and some have been implemented in production vehicles. This section summarizes a number of approaches to rollover detection. A more in-depth review of rollover detection methods can be found in [50].

i. Roll Angle and Roll Rate Measurement

If the vehicle is equipped with sensors capable of measuring the roll angle ϕ and roll rate $\dot{\phi}$, rollover detection can be performed simply by analyzing these measurements. The simplest approach would be to define a threshold value of the roll angle ϕ_{max} and to switch on the controller when$|\phi| > \phi_{max}$. In order to introduce predictive action to the algorithm, the roll rate measurement may be used. For instance, the controller may be switched on when $|\phi| > \phi_{max}$and$\dot{\phi}\, sign(\phi) > 0$. An obvious disadvantage using roll angle and roll rate measurements is that extra sensors are required.

ii. Load Transfer-based Methods

Since load transfer occurs in connection with rollover events, the load transfer ratio is often used for rollover detection as well as control [49]. The case of $LTR = \pm 1$, corresponding to the point at which one wheel begins to lose contact with the road surface, is often used as a 'critical situation' which should be avoided in order to prevent rollover.

iii. Energy-based Methods

The detection of a rollover event, and the subsequent activation of the controller, can be performed by considering the roll energy of the vehicle [51]. The 'critical situation' is defined as when two wheels begin to lift from the ground, that is, the normal forces for these wheels become zero. In this situation, complete load transfer occurs, meaning that the total normal force on the side of the vehicle remaining in contact with the ground is equal to the vehicle's weight mg. It is assumed that the total lateral tire forces F_{yT} for the wheels retaining contact are at their maximum levels, so that $F_{yT} = \mu F_{zT} = \mu mg$. The roll energy associated with the vehicle is composed of a potential energy part involving the energy stored in the suspension springs as well as the height of the center of gravity, and a kinetic energy part. The roll energy is given by:

$$E = \frac{1}{2} C_{\phi} \phi^2 - mgh_{CoG}(1 - \cos\phi) + \frac{1}{2}\left(I_{xx} + m{h_{CoG}}^2\right)\dot{\phi}^2. \tag{2.52}$$

A critical value of the roll energy, $E_{critical}$ can now be found, which represents the minimum possible roll energy in the critical situation of wheel lift-off.

iv. Lateral Acceleration-based Methods

Dynamic rollover threshold can be determined through lateral acceleration [52]. As has been indicated earlier in this section, the root cause of the rollover problem can be seen as the D'Alembert pseudo-force ma_y acting at the CoG and giving rise to a roll moment. Since the lateral acceleration a_y is usually measured in vehicles equipped with measurement systems, its use in rollover detection is attractive. The lateral acceleration limits obtained previously in this section could be used as a starting point for obtaining a switching algorithm based on the lateral acceleration measurement. However, these limits are based on greatly simplified models.

2.4 Model Predictive Control and ACADO Toolkit

2.4.1 Introduction

In this section of the literature review chapter, Model Predictive Control (MPC) and Automatic Control and Dynamic Optimisation (ACADO) toolkit are introduced. The aim is to come up with a control method after reviewing vehicle stability in the previous section. The following literature on MPC is based on a book by Jim Rawlings and David Mayne [53]. Literature on ACADO Toolkit is based [54] and references therein.

2.4.2 Model Predictive Control (MPC)

Model Predictive Control (MPC) has its roots in the field of optimal control. The basic idea of MPC is to use a dynamic model to predict the system's behaviour, and optimize this

behaviour to obtain the best control decision at the current time. Usually, MPC is carried out with a receding horizon: at each time step, the behaviour up until some time horizon is considered. The dynamic models, which are central to any form of MPC, of ordinary differential equation (ODE) are used in this study. The models are of the following form:

$$\frac{d\xi}{dt} = f(\xi, u, t), \tag{2.53}$$

$$y = h(\xi, u, t), \tag{2.54}$$

$$\xi(t_0) = \xi_0. \tag{2.55}$$

In which $\xi \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the input, $y \in \mathbb{R}^p$ is the output and $t \in \mathbb{R}$ is time. The state equations and the output equations are denoted by f and h, respectively. $\mathbb{R}^n$ is used to denote the set of real-valued n-vectors. The initial condition specifies the value of the state ξ at time $t = t_0$, and a solution is searched to the differential equation for time greater than t_0, $t \in \mathbb{R}_{\gg 0}$. Often the initial time is defined to be zero, with a corresponding initial condition, in which case $t \in \mathbb{R}_{\gg 0}$.

MPC tries to minimize a certain cost function, defined as follows:

$$V_T\big(\xi(\cdot), u(\cdot)\big) = \int_0^T l\big(\xi(t), u(t)\big)dt + V_f\big(\xi(T)\big), \tag{2.56}$$

Where $l(\cdot)$ is denoting the running cost and V_f the final cost. The optimal control problem (OCP) formulation of MPC to be solved is then

$$\underset{\xi(\cdot), u(\cdot)}{\text{minimise}}\, V_T\big(\xi(\cdot), u(\cdot)\big), \tag{2.57}$$

$$\text{subject to } \dot{\xi} = f(\xi, u), \qquad t \in [0\ T], \qquad \xi(0) = \xi_0.$$

A first contrast that MPC makes with more traditional controllers is the fact that the control decisions are optimized with respect to a model [53]. For instance, in Proportional Integral Derivative (PID), there are no optimization features (except for the tuning of the gain parameters). Another big advantage of MPC over conventional feedback controllers is the ability to handle constraints, by introducing them in the control problem formulation. A third beneficial property of MPC is that it is model-based, allowing the prediction of future behaviour of the system model, where in other controllers the control action is often based on the tracking error give as:

$$e(t) = y_{ref}(t) - y(t) \tag{2.58}$$

This results in no look-ahead features.

2.4.3 Nonlinear MPC

There are mainly two forms of MPC: linear MPC and nonlinear MPC (NMPC) [53]. In linear MPC, the system equations are assumed to be linear, and only a quadratic cost function is allowed with linear constraints. In nonlinear MPC the system equations are nonlinear. A disadvantage of linear MPC when used in a nonlinear setting is the need for an adequate linearization. This linearized model is only valid close to the linearization point. However, when changing to a nonlinear MPC formulation, instead of solving one Quadratic Program (QP) [55] per iteration, a nonlinear program (NLP) has to be solved. In general, this NLP is non-convex, so there is no guarantee that the solution found by the solver is a global optimum.

The formulation of a generic continuous NMPC problem is as follows:

$$\underset{\xi(.),u(.)}{\text{minimise}} \quad \sum_{0}^{T} l(\xi(t),u(t))dt + V(\xi(T)) \tag{2.59a}$$

$$\text{subject to} \quad \dot{\xi} = f(\xi(t),u(t)), \qquad \forall t \in [0,T], \tag{2.59b}$$

$$h(\xi(t),u(t)) \leq 0, \qquad \forall t \in [0,T], \tag{2.59c}$$

$$\xi_{min} \leq \xi(t) \leq \xi_{max}, \qquad \forall t \in [0,T], \tag{2.59d}$$

$$u_{min} \leq u(t) \leq u_{max}, \qquad \forall t \in [0,T], \tag{2.59e}$$

$$\xi_0 = \xi_{init}. \tag{2.59f}$$

The terms l, V, f, h are arbitrary nonlinear functions representing running cost, terminal state, system state equation and output equation respectively. $\xi \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the input, and $t \in \mathbb{R}$ is time. The fixed length of the prediction horizon is given by T. Maximum and minimum bound (constraints) on states are denoted by ξ_{max} and ξ_{min} respectively. Maximum and minimum bound on control inputs are denoted by u_{max} and u_{min} respectively. The initial condition specifies the value of the state ξ at time $t = t_0$, and a solution is searched to the differential equation for time greater than t_0, $t \in \mathbb{R}_{\gg 0}$.

2.4.4 NMPC Solution Method

To solve the NMPC problem in equation 2.59, numerical optimisation methods are used. The most commonly used numerical optimisation methods to solve OCP's are the Newton-type optimisation methods. They are used in problems where the cost function V_T and the constraints functions are at least twice continuously differentiable. These are *Sequential*

Quadratic Programming (SQP) Method and *Interior Point (IP)* Method. An overview of these methods and others can be found in [56].

i. Sequential Quadratic Programming (SQP) Method

The method is an iterative set-up where the OCP is modelled for a given iterate ξ^k, $k = 0, \dots, N-1$. This results in a Quadratic Programming (QP) sub-problem, the Quadratic Programming (QP) sub-problem is solved and then the solution is used to establish iterate ξ^{k+1}. This is performed in a manner that the solution at each iteration converges to a minimum ξ of the OCP as $k \to \infty$.

ii. Interior Point (IP) Method

For this method, the inequality constraints are changed to a set of equality constraints to accurately approximate the original OCP. Then the Newton's method is applied to the new problem. This method finds the local solution rather than the global solution to the OCP's.

Based on these optimisation methods (SQP and IP), optimisation software packages have been developed to solve the Optimal Control Problems in particular and non-linear programming problem in general. Some of these packages are ACADO, IPOPT, PROPT, dsoa, MUSCOD-II, OptCon and NEWCON. More information on these packages is presented later in this section.

2.4.5 NMPC Stability

There are a number of ways to ensure stability of an NMPC controller. The ways are briefly covered here. More information on these can be found in many books on MPC including [57], [53].

i. Terminal constraints

One way of ensuring stability in NMPC is to have any length of horizon but to add a terminal constraint which forces a state to take a particular value at the end of the prediction horizon. This can be proved by using Lyapunov function, as was first shown by Keerthi and Gilbert [58].

ii. Infinite Horizon NMPC approach

In this approach the prediction horizon is infinity and the cost function is bounded, which implies asymptotic stability of the closed-loop system. A mathematical proof of this is in many MPC books, for example in [53], [57]. It should be mentioned that a direct solution for the infinite horizon problem is impossible and an approximation of the infinite horizon NMPC approach is used instead.

iii. Finite and Quasi-infinite Horizon Approaches

In these two approaches, system states constraints (equality or inequality constraints) are added to the Optimal Control Problem (OCP) in order to secure the closed loop stability of the system. In addition to the states constraints, a further penalty term is added to the cost function. To guarantee closed loop stability, the state constraints to be added to the OCP and the additional penalty term of the cost function must be selected paying closer attention to the open loop system and in such way that the OCP still has a feasible solution set. A description of the Quasi-finite Horizon NMPC approach can be found in [59].

2.4.6 NMPC Robustness

Generally, a robust controller is required to ensure that model mismatch, input and output disturbances, measurement noise and system uncertainties do not affect the expected performance of the closed loop system. The Nonlinear Model Predictive Control is inherently robust enough to handle small size disturbance provided that the dynamic model used for the prediction is accurately described to the physical system [57].

Nevertheless, there are NMPC formulations to assure NMPC controller's robustness. $H\infty$ can be combined with NMPC to ensure robustness. This approach has a high computational time requirement since not one but two optimisations are performed at each sampling period. Another approach to ensure robustness is to soften the hard constraints in the OCP of the NMPC problem. The Stochastic NMPC also presents an approach to deal with disturbances. This approach is based on the occurrence probabilities of the disturbances on the closed loop system. In this approach, the system optimisation is done taking into account all the eventual disturbances that could occur in the system. This approach raises the computation times elapsed on the optimisation.

2.4.7 ACADO Toolkit

Apart from Automatic Control and Dynamic Optimisation (ACADO), a number of optimisation software packages for solving nonlinear optimal control problems are available in literature. Some of the frequently used are as follows:

- ***IPOPT*:** A very popular interior-point algorithm for optimising large-scale differential algebraic systems, IPOPT is an open-source package [60].
- ***PROPT*:** This is Matlab package [61]. It is a commercial software tool developed by Tomlab Optimization Inc.
- ***dsoa*:** The open-source code dsoa was published by Brian Fabien [62].
- ***MUSCOD-II*:** Another SQP-based optimal control package is MUSCOD-II, originally developed by Daniel Leineweber [63].
- ***OptCon and NEWCON***: The open-source packages OptCon [64] and NEWCON [65] both employ a multiple shooting method in combination with the sparse SQP method.

Each of the packages has its particular strengths and all of them have proven successful for a specific range of applications. As they are all tailored to a certain choice of underlying numerical algorithms, it is usually problem-dependent which one is most suited. Moreover, their specialised software design renders it difficult to combine algorithmic ideas from different packages or to extend them with new mathematical concepts. Finally, most packages aim at users with a sound programming background and some of them are either not freely available or rely on proprietary external packages (like the sparse linear system solver within IPOPT).

The ACADO Toolkit (based on SQP), has been designed to overcome these issues. Besides the efficiency of the implementation, four key properties have been identified that are believed to be crucial for any automatic control software based on dynamic optimisation. These properties have guided the implementation of the ACADO Toolkit:

i. *Open-source*: The package is freely available and distributed under the GNU Lesser General Public License (LGPL).
ii. *User friendly*: The syntax for formulating Optimal Control Problems is as intuitive as possible. It makes intensive use of object-oriented capabilities of C++ in particular operation overload. So the formulated problems are as close as possible to the mathematical syntax.
iii. *Extensible code*: ACADO Toolkit facilitates object oriented software design concepts such as abstract base classes and inheritance. In that manner it is easy for the developers to link existing algorithms while avoiding code duplication. Almost all the algorithmic parts can be used stand-alone.
iv. *Self-contained*: The usage of external packages is optional, making the toolkit to run its core features in a stand-alone mode. For example, external packages such as graphical output are optional.

2.4.8 ACADO Problem Classes

The current version (at the writing of this report) of the software supports solvers for four problem classes [54].

- *Optimal Control problems and offline dynamic optimization problems* that aim to solve open loop control inputs, minimizing an objective function with given constrains.
- *Multi-objective optimization* and optimal control problems. Those that aim at simultaneous minimization of more than one objective functions.
- *Parameter and state estimation problems*. Given a dynamic system, the algorithm tries to find unknown control inputs or parameters by measuring the systems output.
- *Model predictive control problems* and online estimation in which dynamic optimization problems are being solved repeatedly.

2.4.9 ACADO Code Generation

The ACADO Toolkit also provide *Code Generation* feature. The ACADO Code Generation is an external tool impended to the ACADO Toolkit. The tool generates real time iteration algorithms for nonlinear MPC. The ACADO Code Generation makes use of the symbolic features of the ACADO Toolkit and exports highly efficient C-code that is tailor made for MPC formulation problems. The idea is to automatically generate the source code of optimization problems for speeding up the computation of numerical solutions. For more details refer to [54].

Chapter 3

Review of Related Works

The aim of this chapter is to present a detailed review of the related works to the problem studied in this research. The chapter also aims at explaining further the problem statement given in the introduction chapter of this report. This presentation comes after literature review of materials that are useful in relation to the problem under consideration; that is mainly rollover prevention and path following of a vehicle.

3.1 Related Works

Rollover detection systems and anti-rollover control systems have been presented by different authors in literature. Different authors approach the rollover prevention problem from different angles. Mostly the differences are based on both detection and control of vehicle rollover. The following are related works to the problem being studied:

- In [18] a model-based roll estimator was designed to estimate the roll angle and roll rate of the vehicle. A rollover index (RI) which indicates an impending rollover was developed by a roll dynamics phase plane analysis. The rollover index was calculated using roll angle, roll rate, lateral acceleration and time to wheel lift (TTWL). A model-based roll estimator estimated using a linearized three degrees-of-freedom bicycle model (including the roll equation) and extended Kalman filter with lateral acceleration, yaw rate, steering angle and vehicle velocity measurements. A rollover mitigation control (RMC) scheme threshold was determined from the rollover index. Differential braking control law was applied to the RMC and the threshold of RMC was designed by RI. The brake control inputs were directly derived from the sliding control law based on a linear bicycle vehicle model with differential braking.

  Simulations were performed using CARSIM and MATLAB. Simulation results illustrated that the proposed RI could be a good measure of the danger of rollover. It was concluded that the RI-based RMC scheme can efficiently reduce the risk of a rollover.

  It is seen from the paper that the authors only considered rollover prevention. Furthermore, differential braking was applied as a control method. This way of preventing rollover makes the vehicle deviate from the intended path. Although the trajectory followed by the vehicle after applying the control mechanism is not displayed in the paper, an extract of a graph showing tire normal forces is shown in figure 3.1. From the figure oscillatory behaviour can be seen in the tire normal forces, which translates into failing to follow a defined yaw rate for a specific trajectory. From the same figure

3.1, it is observed that for example between 2 and 3 seconds (a space of 1 second); there are almost three extreme peak tire force values. This can be viewed as aggressive actions and in real implementations it can require actuators with very fast speed of response. Furthermore, considering that the simulations were performed at a speed of 80 km/h, such actions can easily result in yaw rate instability.

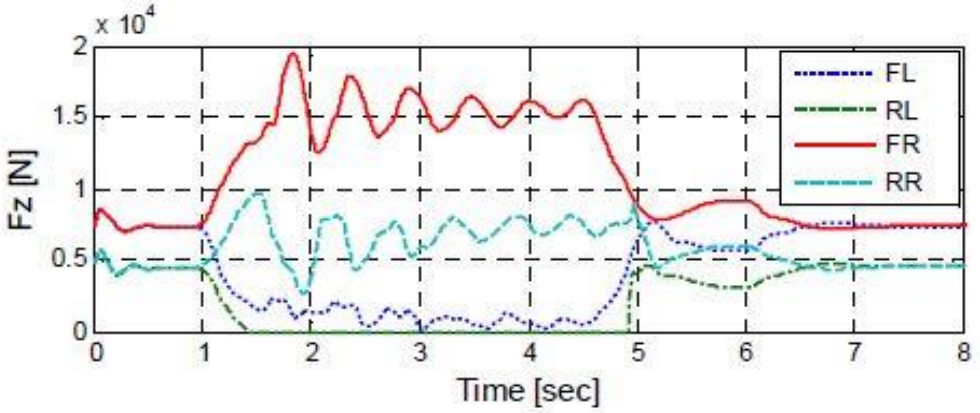

Figure 3.1 Showing front left (FL), front right (FR), rear left (RL) and rear right (RR) tire normal forces [18].

- A similar approach as in [18] is found in [19]. The work in [19] is an improvement of the work in [18]. Similar models and simulation software were used in these works. In [19] a unified chassis control (UCC) strategy to prevent vehicle rollover and improve vehicle lateral stability is described. Specifically, RI/lateral stability-based rollover mitigation (ROM) controller was designed to reduce the danger of rollover without loss of vehicle lateral stability. This was done by integrating electronic stability control (ESC) and continuous damping control (CDC). In terms of comparisons, the RI/lateral stability-based ROM control system showed improved performance compared with the RI-based ROM control system. The better performance was noted from the viewpoint of the lateral stability, such as the yaw rate error and the tracking error. Figure 3.2 shows a snapshot of results from [19] showing yaw rate error for the two controllers.

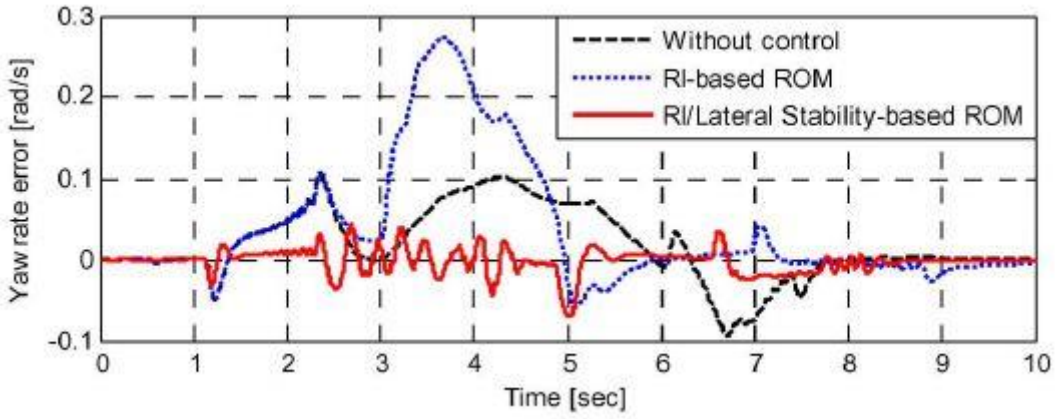

Figure 3.2 Comparison of RI-based ROM and RI/Lateral Stability-based ROM in terms of yaw rate error [18].

In figure 3.2 it is observed that the yaw rate error for the RI/Lateral Stability-based ROM is smaller compared to that of RI-based ROM. This can be viewed as an improvement on the controller in terms of trajectory tracking. The improvement comes in due to the continuous damping mechanism, which is an added cost of course. Other comments made on results in [18] are still not addressed, for example oscillations.

- In [20], [7], [21] another measure of performance, apart from rollover index, for rollover detection is introduced. This measure is called Load Transfer Ratio (LTR) (LTR was described in literature review chapter of this report). Similarly to the rollover prevention mechanisms mentioned; differential braking- based rollover controllers have been designed to keep the value of this quantity below a certain threshold. Specifically in [7], controllers which yield robustness to variations in vehicle speed were obtained. It was noted that the load transfer ratio is related to tire lift-off and it can be considered as an early indicator of impending vehicle rollover. Linear bicycle vehicle model with roll dynamics equation was used. The parameters of the model were tuned against the dynamics of a compact passenger vehicle such that there is a perfect match at steady state.

  The aim of their control strategy was to maximize the magnitude of the allowable disturbance inputs which do not drive the performance outputs outside their specified limits; the disturbance input was considered as the driver steering input. Figures 3.3 and 3.4 show control force plotted against time and global position of vehicle centre of gravity respectively. Shown on the graphs are two designed controllers named fixed model controller and robust controller. Additionally, uncontrolled vehicle model path is shown on figure 3.4 as well.

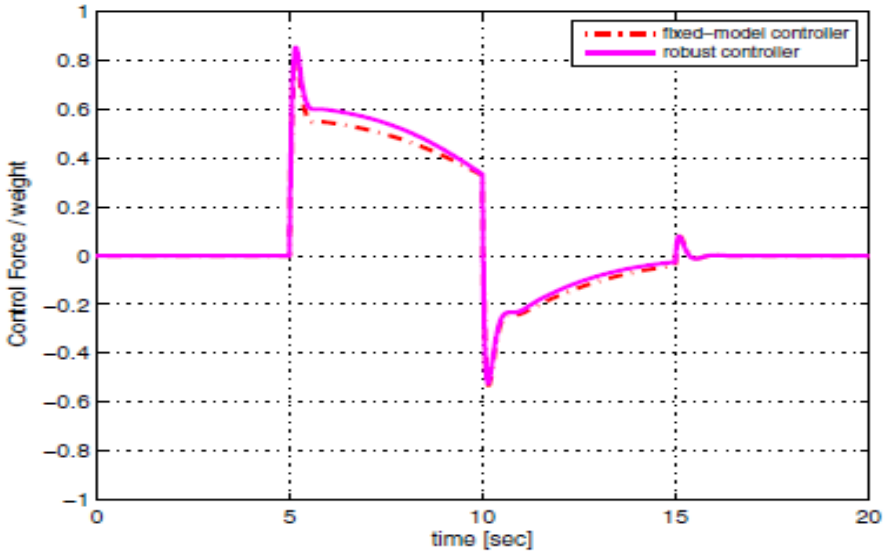

Figure 3.3: Control force for fixed model controller and robust controller

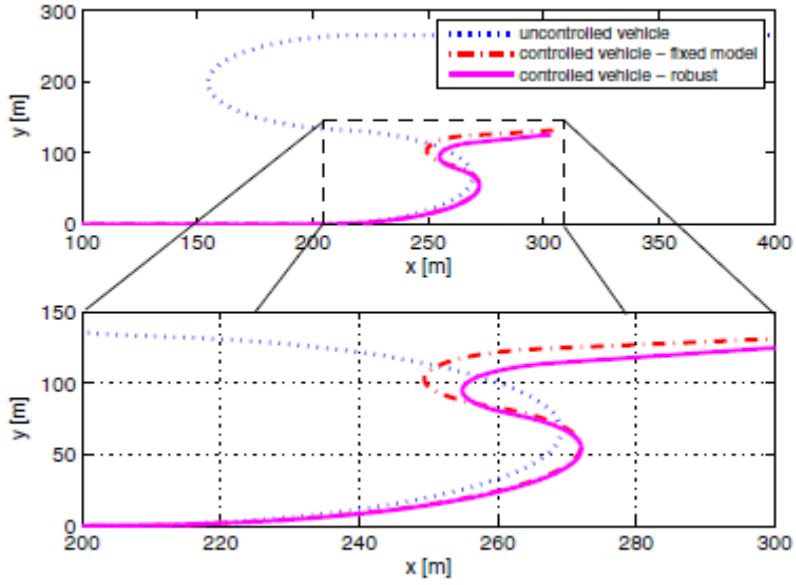

Figure 3.4: Showing global position of the controlled vehicle and uncontrolled vehicle [7].

It is observed from figure 3.4 that rollover prevention controllers resulted in the vehicle deviating from the planned trajectory. The planned trajectory is represented by the uncontrolled vehicle graph. It is also observed in figure 3.3 that the control forces change abruptly (aggressive actions), this was during the times when the threshold LTR was reached.

Many of the rollover detection and prevention schemes act when vehicle instability is imminent, which results in aggressive actions [22]. In addition to aggressive actions, they

need the vehicle to be equipped with actuators with fast speed of response. This can easily be observed in the related works presented in the previous paragraphs. A common framework in the literature which can solve these problems is model predictive control (MPC).

MPC theory has been applied to a variety of process control applications [23]. Due to high computational power needed, only slow plants used to benefit from MPC. However, everyday progress in processors speed and new off-line techniques [24] together with the ability of input, states, and output constraints handling make MPC much more applicable to ordinary and fast plants. MPC is the only control technology that can systematically take into account the future predictions and safe system operating limits (constraints) in the design stage [25]. This makes it a suitable choice for systems where the system faces dynamically changing environment and has to satisfy crucial safety constraints (such as obstacle avoidance) as well as actuator constraints.

Prevention of vehicle rollover using model predictive control is reported in literature as well. Examples of rollover prevention using MPC are found in many publications including [26], [27], [28], [29] and [22].

- In [26], a linear model predictive control (MPC) approach to roll stability of a scaled crash avoidance vehicle was presented. A roll stability controller (RSC) based on an eight degree of freedom dynamic vehicle model was designed. A linear Pacejka tire model was used. System identification was done by disassembling a test platform vehicle and measuring the components dimensions and masses, then building a solid works model of the same vehicle. Parameters which could not be measured directly were then identified from the solid works model. The controller was designed for and tested on a scaled vehicle performing obstacle avoidance manoeuvres on a populated test track. A rapidly-exploring random tree (RRT) algorithm was used for the vehicle to execute a trajectory around an obstacle. The vehicle examined the geographic, non-homonymic, and dynamic constraints to manoeuvre around the obstacle.

  A model predictive controller (MPC) used information about the vehicle state and, based on a weighted performance measure, generated an optimal trajectory around the obstacle. An emphasis was placed on the mitigation of rollover and spin-out. Both MPC and RRT were run on MATLAB. The results demonstrated a markedly improved stability and manoeuvrability for double lane changes, simple turns, and RRT path planning at speeds up to 3 m/s and turn angles of up to 20 degrees on a 1/10th scale vehicle. Figure 3.5 shows steering angle control efforts of the proposed MPC controller compared with a proportional derivative (PD) controller extracted from the paper.

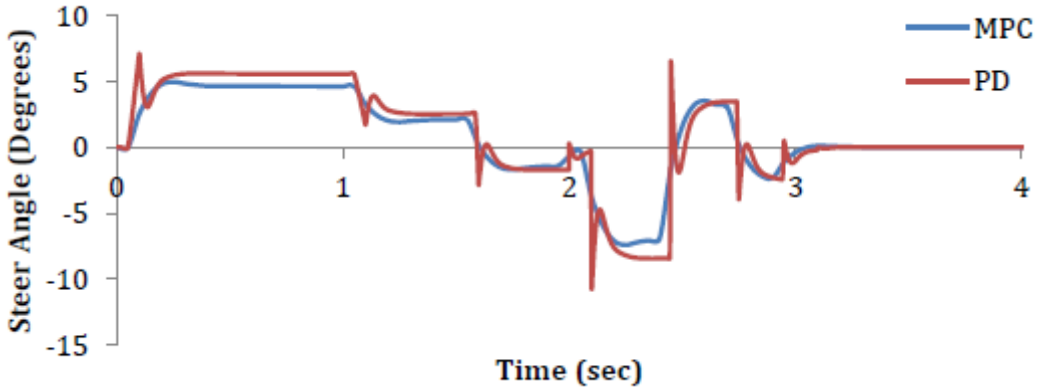

Figure 3.5: Comparison of MPC and PD control efforts [26].

From figure 3.5, the MPC is controller is seen having smoother control efforts as compared to PD controller. This performance is achieved by the MPC ability to predict future response of the system as well as allowing constraints.

The author considered roll angle and spin out stability during obstacle avoidance only. Roll angle and spin out stability can also occur during for example following a curved road as explained earlier on. Trajectory tracking after obstacle avoidance was not treated, this can result the vehicle bumping into other vehicles or obstacles after avoiding another.

- In [27], linear model predictive control (MPC) was used within a multilevel control framework; the lower level controller (LLC) consisted of two subsections: predictive steering controller and brake controller. Lower level model predictive steering controller predicted the vehicle condition during next time horizon. The prediction was based on linearized vehicle model and tire test data was used. Tire test data was similar to that generated by Pacejka tire model described in literature review chapter. The lower level predictive steering controller intervened in driver's command to avoid an imminent rollover. Losing the driver's desired path was the penalty however.

  MPC of steering confined the vehicle rollover risk within the predefined limit. Rollover prevention was done online by the lower level controller. Load transfer ratio (LTR) index was used to detect rollover event. Proportional Derivative (PD) brake controller of LLC slowed down the vehicle to the desired speed of the upper level controller (ULC). The ULC desired speed was estimated from an algorithm. Simulation results showed that the proposed control framework maintained roll stability while tracking error was confined to predefined error limit. Simulations were performed in MATLAB.

  The aim of the paper was to prevent rollover and at the same time to reduce trajectory tracking error. Since MPC was included in multilevel control structure, it resulted in MPC

performing poorly if left alone to control the vehicle. Looking at the strength of MPC (nonlinear MPC in this case), the framework can be capable of achieving both path following and rollover prevention without being added on to other controllers in multilevel structure.

- In [28], linear model predictive control (MPC) scheme was proposed that potentially could achieve both yaw stability and roll stability. The objective of the proposed MPC algorithm was to track a reference yaw rate and body side slip angle, which served the purpose of yaw stability control. Moreover, a rollover prevention constraint was imposed to achieve roll stability. The MPC optimized the linearized three degrees of freedom vehicle dynamics model by stabilizing the yaw motion while observing the rollover prevention constraint (limited lateral acceleration), hence the yaw and roll stability could be achieved simultaneously.

  A high fidelity heavy-duty truck model from TruckSim was used for simulations. Pacejka tire model was used whose parameters were fitted using the least squares approach applied to TruckSim tire data file. The effectiveness of the proposed approach was demonstrated by the TruckSim and Simulink joint simulations. It was found that the optimization results from MPC intentionally introduced larger slip angles into the system to avoid rollover as can be seen in figure 3.6 extracted from the paper.

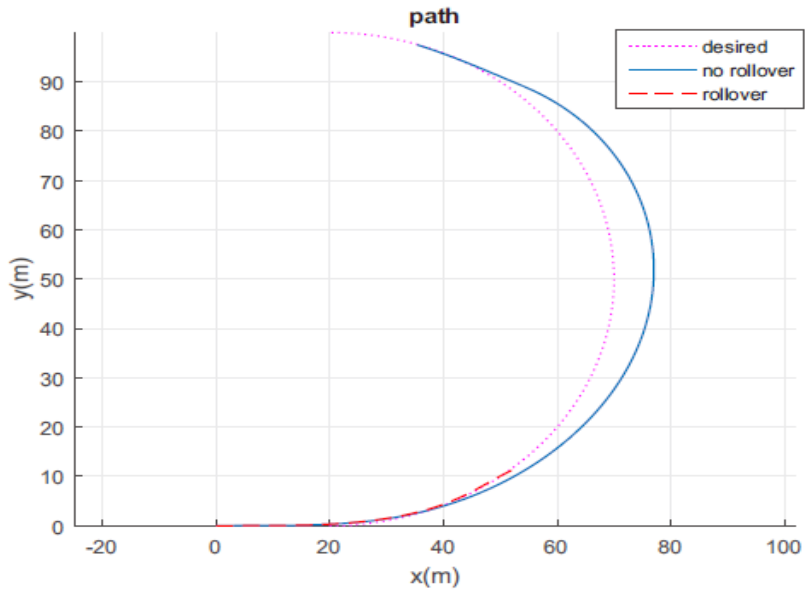

Figure 3.6: Global path of the vehicle with desired path [28].

From figure 3.6, it is observed that trajectory tracking error is large during rollover prevention. Although the authors applied full MPC in this paper (not within multilevel structure as in [27] discussed in previous paragraphs), they still used differential breaking, hence the large tracking error seen in figure 3.6. The differential brakes were applied to

achieve tracking the reference yaw rate at the same time respecting the lateral acceleration constraint (rollover prevention) without actually taking into account the path being tracked. This made the controller fail to satisfy its intended design goals.

- In [29], a control method of switching the linear MPC controllers which uses differential braking with dead zone and active rear steer (ARS) was proposed. A single truck model (bicycle model) with roll equation was used in conjunction with linear tire model (Pacejka). The aim was to prevent rollover and remove trajectory tracking error. Load transfer ratio (LTR) was used as a rollover event indicator.

  A typical obstacle avoidance manoeuvre was simulated with a peak driver steering input of magnitude equal to 5.8 degrees at the wheels. Priority was put on rollover mitigation through differential breaking, after which ARS controller was switched on to track the driver's intended trajectory. Switching time depended on duration taken by the driver to complete obstacle avoidance, and was set to 4 seconds during the simulation. Simultaneously, the trade-off between rollover prevention and path tracking was highlighted through simulation results. The effectiveness of using switching controllers designed for the trade-off solution was also confirmed through MATLAB/Simulink simulation results. Figure 3.7 shows a graph of vehicle global position and the reference trajectory.

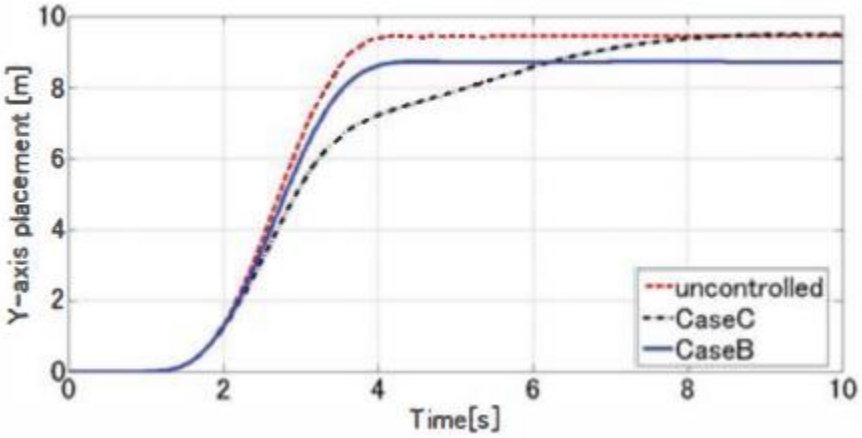

Figure 3.7: Showing reference trajectory (labelled as uncontrolled) and vehicle trajectory with the implemented MPC controller with ARS (labelled as Case C) [29].

In the extracted figure 3.7, the graphs of interest are the reference trajectory (labelled as uncontrolled) and the trajectory followed by the vehicle after the explained MPC with ARS was implemented (labelled as Case C). The third graph labelled as Case C is of no importance in this study, hence should be ignored.

Looking at the vehicle global position shown in figure 3.7, it is clearly seen that the vehicle failed to tack the trajectory by a huge error. The error can mainly be seen to be between approximately 3 seconds and 7 seconds. This is the time the controller was preventing rollover event by differential braking. After that it is when ARS was activated to remove the tracking error. Although the error is seen to be removed after activating ARS, but in practical implementation the time taken to activate ARS can result in poor performance of the controller. The aim of following driver's trajectory is to avoid causing other accidents. Therefore, this controller does not necessarily achieve the purpose, because an accident would have happened already during the huge error observed before ARS acted.

- In [22] a control algorithm to follow a curved road while simultaneously preventing rollover was introduced. Model predictive control was applied to minimize roll motion throughout cornering. The prediction of vehicle state was based on a four-wheel nonlinear vehicle model with roll dynamics and a brush tire model. Full braking was utilized as a control actuator to achieve an optimal balance in the trade-off between the vehicle-speed and roll motion.

  To come up with a control law, the vehicle states were estimated using Kalman Filter adapted for nonlinear systems, called an Unscented Kalman Filter. The proposed controller then fed the desired velocity into the low-level speed controller. During simulations, done in CarSim, tests were conducted which involved the MPC control law working with the low level speed control law (labeled Ctrl in figures 3.5 and 3.6), and the low level speed control law only (labeled No Ctrl in figures 3.5 and 3.6). The region shaded with vertical lines corresponds to the time period that the controller was activated. It was found that the low level speed control law only did not reduce velocity (figure 3.8) when cornering which resulted into the vehicle deviating from the reference path by around 10 m as can be seen in figure 3.9. The vehicle almost lost stability which was illustrated by the large steering angle to compensate for the deviation. The maintained speed was 105 km/h at a turning radius of 100 m.

  When the MPC controller was tested, the vehicle velocity reduced to the optimal velocity, about 94 km/h after entering the curve. Consequently, the vehicle did not lose stability, and this was illustrated by the small steering wheel and roll angles.

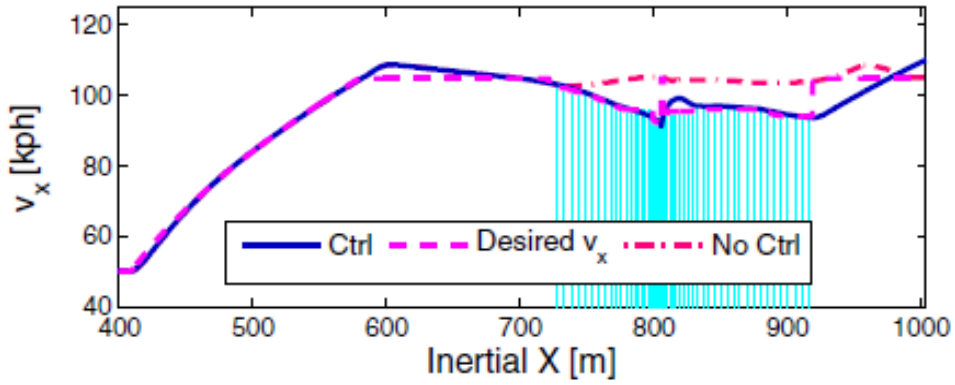

Figure 3.8: Showing longitudinal velocities of the vehicle [22].

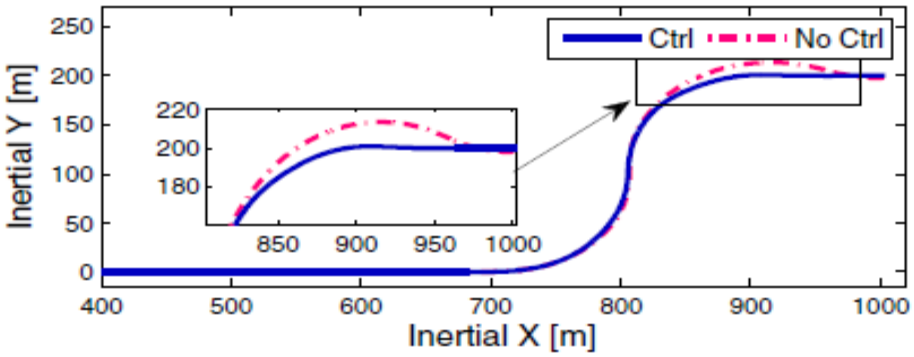

Figure 3.9: Showing global position of the vehicle [22].

From these results in figures 3.8 and 3.9 it is observed that the use of MPC has removed the aggressiveness of control actions. This can be seen in the velocity graphs (figure 3.8) where the velocity is seen reducing gently during vehicle cornering. Furthermore, the trajectory tracking error is seen tremendously reduced in figure 3.9. The reduction in trajectory error is as a result of combination of MPC and full braking as a control actuation.

The MPC controller discussed so far whose results are seen in figures 3.8 and 3.9 was activated only when the vehicle was cornering. This criterion assumes that rollover event can only happen when vehicles are cornering. In practical applications, rollover can occur under different situations as highlighted in the introduction chapter of this report.

3.2 Detailed Research Problem

In the first part of this chapter in related works section, it is noted that many of the rollover detection and prevention schemes act when vehicle instability is imminent, which results in aggressive actions. In addition to aggressive actions, they need the vehicle to be equipped with actuators with very fast speed of response, which is a challenge sometimes in practical

situations. As a way of trying to solve these problems, the chapter introduced an advanced control architecture known as model predicative control (MPC) theory.

MPC has been applied in a number of publications tackling the rollover prevention and trajectory following problem. A number of such applications have been discussed in the second part of this of this chapter. From the examples provided, it is seen that most of the researchers put much emphasis on rollover prevention and less emphasis on trajectory tracking. This has led to successful rollover prevention while trajectory tracking is affected and not thoroughly investigated. In most of the publications, differential braking is used as a control actuator to prevent rollover. Differential braking contributes to poor trajectory tracking because it precludes required yaw rate to follow a curved road. Not tracking intended (defined) trajectory during rollover prevention can cause other accidents as well, for example hitting obstacles or other vehicles. This is crucial in both manned and un-manned vehicles.

Furthermore, it has been noted that most authors use linearized models during rollover detection and prevention. Linearization of models which leads to linear MPC improves computation speed. The problem of linearizing highly nonlinear models is that linearized models perform better when operated close to the point of linearization. Performance tends to be poor if operated far from the point of linearization. This point has not been investigated (mentioned) by most of the authors who have used linear MPC in controlling vehicle rollover event.

The use of linear model is in MPC to represent nonlinear systems is mainly to reduce computation intensity. With the advancements in microprocessor technology as indicated in previous sections, adopting nonlinear MPC for nonlinear systems seems viable. Nonlinear MPC was used in [22] as discussed in previous paragraphs. In [22] both rollover prevention and path following were treated. The treatment of both were considered only when the vehicle was cornering.

Considering that rollover can occur at any point, the work in this study extends the idea in [22] by applying nonlinear MPC to the whole path of travel of an autonomous vehicle. In addition to constraining roll angle; yaw angle, yaw rate and inputs are constrained as well. Full braking aimed to achieve the yaw rate required to follow a curved path while simultaneously preventing rollover is used. Most of the related works presented in this chapter used commercial software for example MATLAB, CarSim and TruckSim during their study. In this study, open source software (OSS) are used. Specifically, Scilab and ACADO are used. Use of OSS provides flexibility to controller design by allowing changes to software to fit the way you want it be. In addition to being flexible, the software used in this study are freely available.

Chapter 4

Vehicle Dynamics Modelling

This chapter is devoted to modelling of the vehicle dynamics that has been used in this study. A review of mostly used vehicle models has been covered in chapter 2 (literature review) of this study. The work covered here comprises the work covered in [39] and [26].

Almost every introductory vehicle dynamics textbook presents what is commonly referred to as the bicycle model as is shown in literature review chapter under vehicle models section. Developing the bicycle model requires several simplifying assumptions. First, the steering angles for the front wheels are assumed to be equal. It is also required that both the lateral and longitudinal forces induced by the tires are modelled linearly. The linearization is only valid for small steering angles at low acceleration rates. To simplify the model even further, the tire forces occurring at each axle are taken as an average neglecting any effects of dynamically induced weight transfer. By neglecting the body roll the roll-yaw coupling inherent in the general moment equations is lost. As a result the model is only valid for the mildest vehicle manoeuvres. In terms of accident avoidance this is a serious problem because most emergency manoeuvres require the use of extreme turning and braking and the nonlinearities will become extremely important in developing an accurate solution. It is obvious that a more complex model is required.

A far more accurate model would incorporate all of the available degrees of freedom in the vehicle, as well as any and all non-linear tire effects. Because there are thousands of moving parts on a typical vehicle, trying to solve for them all will quickly become physically impossible. By lumping the masses as much as possible (sprung mass/unsprung masses) the model is still accurate but with a much smaller number of degrees of freedom. These include the three degrees of translational and rotational freedom given to the body, as well as the minimum number needed for an accurate suspension model. Such a model (14 degrees of freedom (DOF)) is derived in [39] and it is similar to the one used in commercial software like CarSim. This 14 DOF model includes all three translational and rotational degrees of freedom given to the vehicle body, as well as one rotational and one translational degree of freedom for each tire.

The challenge with these models is that they require an extensive amount of suspension design knowledge before the motions of the tires can be accurately solved. Because the tires are essentially modelled as nonlinear dampers, any inaccuracies in the suspension model will result in incorrect tire velocities, and the tire forces will not be accurate. Use of these models requires the knowledge of suspension link mounting positions and the various degrees of translation/rotational freedom for each link. A model that is more accurate than the bicycle model, but simpler than the 14 DOF would be desirable.

One way to improve the bicycle model is to no longer average the tire normal force and resulting lateral and longitudinal forces. Pitch, roll, and body bounce are still neglected, but now there are four contact points supporting the vehicle mass and the dynamic weight transfer forces are taken into consideration commonly referred to as the track model. The track model with non-linear tires serves as a very solid improvement, but the pitch-roll-yaw coupling is still neglected. Realizing that almost every vehicle suspension carries left-right symmetry there is the possibility to include roll. The roll axis of any vehicle is the line about which the body rolls such that the tires do not have any induced lateral velocities. For some passenger sedans, the roll axis is assumed horizontal. It is then possible to keep track of the roll degree of freedom.

It should be noted that most vehicles lack front-rear symmetries in their suspensions due to anti-pitch and anti-lift mechanisms, and thus determining the pitch becomes very difficult. To simplify the model pitch is neglected, and equations of motion will only include the roll-yaw coupling. The vertical translational degree of freedom is also neglected, which means that the model is only valid for flat, smooth roads. The wheel translational degree of freedom is also neglected, but the rotational degrees of freedom are kept. Additionally, the cornering force is what primarily affects the lateral force on the vehicle, and therefore is the main contributor to rolling and skid-out in a turn. The longitudinal force meanwhile is a factor which plays a role during a braking manoeuvre. The final model is an eight degree of freedom (8 DOF) that covers longitudinal translation, lateral translation, yaw, roll, and the rotation of the four wheels. The list of assumptions is provided as a summary as follows:

- Longitudinal and lateral velocities.
- Roll about x-axis.
- Yaw about z-axis.
- Uses smooth, flat road.
- Lateral tire forces.
- Inputs: current states, steering, braking/driving torque.
- Outputs: Linear/Angular positions, velocities, and rotations and slip angles.

4.1 Coordinate System Definition

In order to accurately model the vehicle dynamics, three sets of coordinate axes are necessary. The method recommended in SAE J670e is adopted. First, the Earth-fixed coordinate axes XYZ are defined and the uppercase letters are used to denote this system. These axes are considered an inertial frame of reference; the coordinates are orthogonal and right handed. A point of origin must be defined as well as the direction of the X axis.

The vehicle-fixed, chassis coordinate system xyz and body coordinate system $x'y'z'$ are located at the same point specified on the vehicle at rest. Assuming the vehicle has lateral symmetry, which is reasonable for most passenger sedans, it is possible to determine the

location of the roll axis. This is the axis that allows the body to rotate without any induced velocities in the tires. By projecting a line vertically downward through the vehicle centre of gravity, the origin of each system is located at the intersection of this line and the roll axis. The advantage of selecting this location is that it provides the simplest means of developing the tire forces. The x and x' axes are in the longitudinal (forward) direction, the y and y' axes are in the right hand lateral direction, and the z and z' axes point vertically downward. The unit vectors for each frame are given as:

i. XYZ $\quad \boldsymbol{n_x}, \boldsymbol{n_y}, \boldsymbol{n_z}$.

ii. xyz $\quad \boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$.

iii. $x'y'z'$ $\quad \boldsymbol{i}', \boldsymbol{j}', \boldsymbol{k}'$.

The unit vectors are listed in $x - y - z$ order. The standard SAE definition of right-hand rotations, starting with xyz aligned with XYZ, are given by:

i. Yaw rotation ψ about the *z*-axis.
ii. Pitch rotation θ about the *y*-axis.
iii. Roll rotation ϕ about the *x*-axis.

These are taken about the vehicle-fixed axes *xyz*. The $x'y'z'$ axis system will roll about the *x*-axis and remain fixed to the body. The transform between $x'y'z'$ and $oxyz$ is used when developing the equations involving the body (see figure 4.1). To allow for compact notation $\star$ denotes the front and rear of the vehicle, while $\bullet$ denotes the left and right side of the vehicle. This notation eliminates the need for redundant equations for the front left, front right, rear left, and rear right segments of the vehicle.

4.2 Linear Equations of Motion

To begin, the origin of the inertial reference frame is defined. The distance from this origin to the origin of the chassis reference frame is:

$$R_o = Xn_x + Yn_y. \tag{4.1}$$

Where X is the x-axis coordinate, Y is the y-axis coordinate, and R_o is the radius with respect to the inertial frame as shown in figure 4.1.

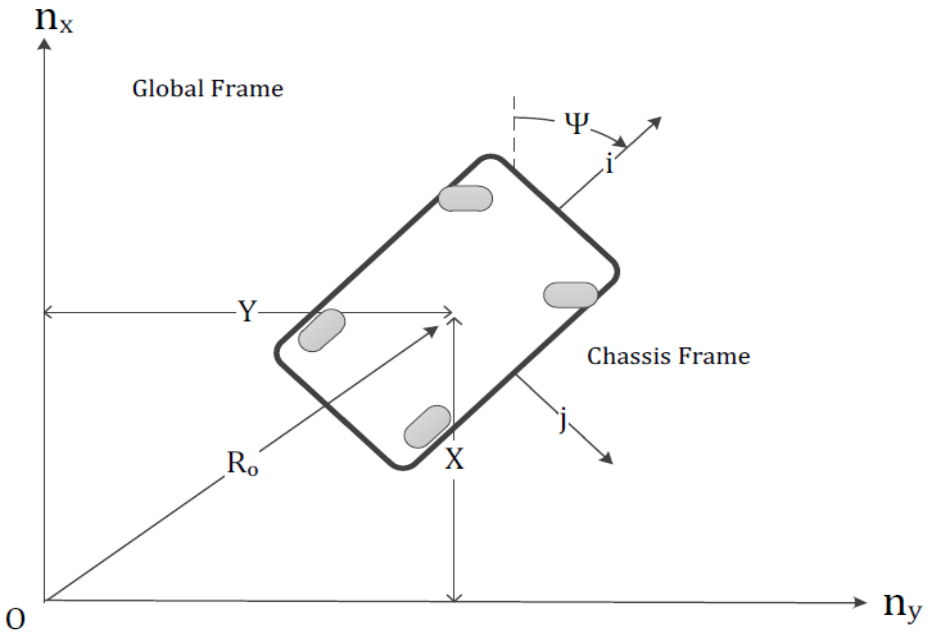

Figure 4.1: Global and chassis coordinate frames [26].

Equation 4.1 is transformed into the chassis frame by setting

$$n_x = \cos\psi i - \sin\psi j, \tag{4.2}$$

$$n_y = \sin\psi i + \cos\psi j. \tag{4.3}$$

Then by inserting equation 4.2 and 4.3 into equation 4.1 it is shown that the radius is equal to:

$$R_o = (\mathrm{X}\cos\psi + \mathrm{Y}\sin\psi)i + (-X\sin\psi + Y\cos\psi)j. \tag{4.4}$$

This radius is given with respect to the chassis unit vectors. The velocity of the chassis origin is given by taking the time derivative of equation 4.4 giving:

$$U_o = \frac{dR_o}{dt} = \dot{X}n_x + \dot{Y}n_y,$$
$$U_o = (\dot{\mathrm{X}}\cos\psi + \dot{\mathrm{Y}}\sin\psi)i + \left(-\dot{X}\sin\psi + \dot{Y}\cos\psi\right)j. \tag{4.5}$$

And by defining

$$U_{ox} \stackrel{\text{def}}{=} \dot{\mathrm{X}}\cos\psi + \dot{\mathrm{Y}}\sin\psi, \tag{4.6}$$

$$U_{oy} \stackrel{\text{def}}{=} -\dot{X}\sin\psi + \dot{Y}\cos\psi, \tag{4.7}$$

the velocity of equation 4.5 is simplified into

$$U_o = U_{ox}i + U_{oy}j. \tag{4.8}$$

This velocity is given in the chassis reference frame. Because the body and chassis systems share the same origin, equation 4.8 defines the linear velocity of the body reference frame as well. To calculate the acceleration of the chassis/body origin the derivative with respect to time is applied again to equation 4.8:

$$a_o = \frac{d^2R_o}{dt^2} = \frac{\partial U_o}{\partial t} + \Omega_c \times U_o. \tag{4.9}$$

To define the angular velocities the fact that the body is free to roll with respect to the chassis must be taken into consideration. Therefore the angular velocity of each is given by:

$$\Omega_c = \dot{\psi}k, \tag{4.10}$$

$$\Omega_b = \dot{\phi}i + \dot{\psi}k, \tag{4.11}$$

and by inserting equations 4.10 and 4.11, the acceleration of the chassis origin is shown to be:

$$a_o = \left(\dot{U}_{ox} - \dot{\psi}U_{oy}\right)i + \left(\dot{U}_{oy} + \dot{\psi}U_{ox}\right)j. \tag{4.12}$$

Where the symbol Ω represents the rotation vector and the subscripts *c* and *b* represent the chassis and body respectively.

With the equation of motion of the chassis coordinate system defined, the equations of motion of the sprung and unsprung masses are considered. The locations of the sprung and unsprung masses are displayed in figure 4.2.

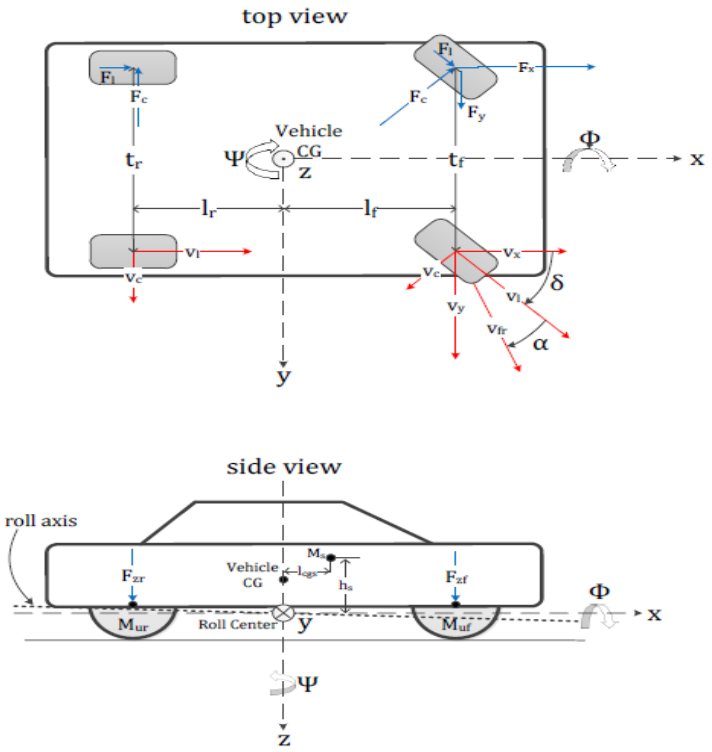

Figure 4.2: Vehicle frame with pertinent vehicle parameters, forces, and velocities [26].

The sprung mass is defined as all parts of the vehicle which are supported by the suspension. The unsprung mass, accordingly, includes the suspension linkages, shocks, wheels, bearings, and brakes. The position of the front unsprung mass is determined from:

$$R_{uf} = R_o + r_{uf}, \tag{4.13}$$

where r_{uf} is the radius in the chassis frame. From the position of the front unsprung mass, in the chassis coordinate system is shown to be:

$$r_{uf} = l_f i - h_{uf} k, \tag{4.14}$$

where l_f is the longitudinal distance between the centre of the front tire and the vehicle centre of mass, and h_{uf} is the height, typically equal to the rolling radius. Now the time derivative of equation 4.14 is taken to get the front unsprung mass velocity:

$$U_{uf} = \frac{dR_{uf}}{dt} = U_o + \frac{\partial r_{uf}}{\partial t} + \Omega_c \times r_{uf}, \tag{4.15}$$

and by inserting equations 4.13 and 4.14 it is shown that:

$$U_{uf} = U_{ox} i + \left(U_{oy} + l_f \dot{\psi}\right) j, \tag{4.16}$$

in the chassis frame of reference. The acceleration of the front unsprung mass is derived by taking the time derivative of the equation 4.16 as:

$$a_{uf} = \frac{dU_{uf}}{dt} = a_o + \dot{\Omega}_c \times r_{uf} + \Omega_c \times \left(\Omega_c \times r_{uf}\right). \tag{4.17}$$

By inserting equations 4.12 and 4.14 it is shown that:

$$a_{uf} = (\dot{U}_{ox} - \dot{\psi}U_{oy} - l_f\,\dot{\psi}^2)i + (\dot{U}_{oy} + \dot{\psi}U_{ox} + l_f\,\ddot{\psi})j, \tag{4.18}$$

in the chassis frame of reference. The exact same process is carried out for the rear unsprung mass. The only change is in the position vector of the rear unsprung mass in the chassis coordinate system. The sequence is presented as follows:

$$R_{ur} = R_o + r_{ur}, \tag{4.19}$$

$$r_{ur} = -l_r i - h_{ur} k, \tag{4.20}$$

$$U_{ur} = \frac{dR_{ur}}{dt} = U_o + \frac{\partial r_{ur}}{\partial t} + \Omega_c \times r_{ur}, \tag{4.21}$$

$$U_{ur} = U_{ox} i + \left(U_{oy} - l_r\,\dot{\psi}\right)j, \tag{4.22}$$

$$a_{ur} = \frac{dU_{ur}}{dt} = a_o + \dot{\Omega}_c \times r_{ur} + \Omega_c \times (\Omega_c \times r_{ur}), \tag{4.23}$$

$$a_{ur} = (\dot{U}_{ox} - \dot{\psi}U_{oy} + l_r\,\dot{\psi}^2)i + (\dot{U}_{oy} + \dot{\psi}U_{ox} - l_r\,\ddot{\psi})j. \tag{4.24}$$

It is now necessary to define the linear motion of the sprung mass. The process is similar to that carried out for the unsprung masses with the exception that the sprung mass is free to roll about the roll axis. Therefore it is necessary to project the sprung mass position vector from the body coordinate frame into the chassis coordinate frame. The vector locating the body is shown to be:

$$R_s = R_o + r_s, \tag{4.25}$$

where r_s is the vector from chassis origin to sprung mass, and R_s is the net position vector. The conversion from body reference frame to chassis reference frame is given by:

$$\begin{aligned} i' &= i, \\ j' &= \cos\phi\; j + \sin\phi\; k, \\ k' &= -\sin\phi j + \cos\phi k, \end{aligned}$$

which are used to redefine the position vector of the sprung mass in the chassis coordinate frame. These equations are shown to give:

$$\begin{aligned} r_s &= l_{cgs} i' - h_s k', \\ r_s &= l_{cgs} i + h_s \sin\phi\; j - \cos\phi\, k, \end{aligned} \tag{4.26}$$

where l_{cgs} is the longitudinal location of the body centre of gravity and h_s is the height with the vehicle at rest. The velocity is determined the same way as before and is shown to be:

$$U_s = \frac{dR_s}{dt} = U_o + \frac{\partial r_s}{\partial t} + \Omega_c \times r_s. \tag{4.27}$$

And by again inserting equations 4.8 and 4.12 it is shown that:

$$U_s = (U_{ox} - h_s \dot{\psi} \sin\phi) i + (U_{oy} + h_s \dot{\phi} \cos\phi + l_{cgs}\, \dot{\psi}) j + h_s \dot{\phi} \sin\phi\, k. \tag{4.28}$$

Due to the relative motion of the body with respect to the chassis coordinate frame, the acceleration of the sprung mass is slightly different from the acceleration of the unsprung masses, and is given by:

$$a_s = \frac{dU_s}{dt} = a_o + \dot{\Omega}_c \times r_s + \Omega_c \times (\Omega_c \times r_s) + 2\Omega_c \times \frac{\partial r_s}{\partial t} + \frac{\partial^2 r_s}{\partial t^2}. \tag{4.29}$$

And by substituting in equations 4.12 and 4.14 it is shown that:

$$\begin{aligned} a_s = (\dot{U}_{ox} - \dot{\psi} U_{oy} - 2h_s \dot{\psi}\dot{\phi} \cos\phi - h_s \ddot{\psi} \sin\phi - l_{cgs}\, \dot{\psi}^2) i + (\dot{U}_{ox} + \dot{\psi} U_{oy} \\ + h_s \ddot{\phi} \cos\phi - h_s \dot{\phi}^2 \sin\phi + l_{cgs}\, \ddot{\psi} - h_s \dot{\psi}^2 \sin\phi) j + (h_s \ddot{\phi} \sin\phi \\ + h_s \dot{\phi}^2 \cos\phi) k. \end{aligned} \tag{4.30}$$

Now by recalling from Newton that:

$$\Sigma F = \frac{dP}{dt} = M a_o, \tag{4.31}$$

the accelerations of the three masses can be summed and rearranged to obtain:

$$M \dot{U}_{ox} = \Sigma F_x + M_s \left(-2h_s \dot{\psi}\dot{\phi} \cos\phi - h_s \ddot{\psi} \sin\phi\right) + M \dot{\psi} U_{oy}, \tag{4.32}$$

$$M \dot{U}_{oy} = \Sigma F_y - M_s \left(-h_s \ddot{\phi} \cos\phi + h_s \dot{\phi}^2 \sin\phi + h_s \dot{\psi}^2 \sin\phi\right) - M \dot{\psi} U_{ox}. \tag{4.33}$$

These are the equations of motion in the longitudinal and lateral coordinates. Noting that the first term in each equation represent the accelerations derived in the *chassis coordinate frame*, while the second term represents the normal acceleration of the chassis frame as it rotates, it is possible to adjust the frame of reference used. Moving the second term from the right side to the left is the equivalent of adopting the rotating frame as the frame of reference. Since this is the chassis frame, and the driver is most familiar with the chassis

frame, it is adopted for the model. The equations 4.32 and 4.33 are therefore modified to obtain:

$$M\dot{U}_{ox} = \Sigma F_x + M_s\left(-2h_s\dot{\psi}\dot{\phi}\cos\phi - h_s\ddot{\psi}\sin\phi\right), \tag{4.34}$$

$$M\dot{U}_{oy} = \Sigma F_y - M_s\left(-h_s\ddot{\phi}\cos\phi + h_s\dot{\phi}^2\sin\phi + h_s\dot{\psi}^2\sin\phi\right). \tag{4.35}$$

The sum of the forces in the x and y directions are given by:

$$\Sigma F_x = F_{xlf} + F_{xrf} + F_{xlr} + F_{xrr}, \tag{4.36}$$

$$\Sigma F_y = F_{ylf} + F_{yrf} + F_{ylr} + F_{yrr}. \tag{4.37}$$

The forces represent the resulting longitudinal and lateral forces provided by the tires. Additional forces, such as aerodynamic drag, can be added as desired.

4.3 Angular Equations of Motion

To define the angular motion of the vehicle the sprung mass angular momentum is defined first. Because the unsprung masses are only permitted to rotate about the *z*-axis, their equations of motion are much simpler and are added later. For the sprung mass the standard definition of the angular momentum is given by:

$$H_c = I_c\Omega_c, \tag{4.38}$$

where H is the angular momentum and I is the inertia tensor. It should be recalled that the rotation vector is given in the chassis coordinate frame, but the inertia tensor is defined for the sprung body in the body coordinate frame. Therefore some details must be given for a proper definition. For the sprung mass, the inertia tensor is given as:

$$I_{sb} = \begin{bmatrix} I_{xxs} & 0 & I_{xzs} \\ 0 & I_{yys} & 0 \\ I_{zxs} & 0 & I_{zzs} \end{bmatrix} + \begin{bmatrix} M_s h_s^2 & 0 & M_s h_s l_{cgs} \\ 0 & M_s(h_s^2 + l_{cgs}^2) & 0 \\ M_s h_s l_{cgs} & 0 & M_s l_{cgs}^2 \end{bmatrix}, \tag{4.39}$$

where the subscript *s* indicates that it is the sprung body and the subscript *b* indicates that this is calculated in the body frame of reference. The first term represents the moment of inertia tensor as calculated about the body center of mass. The zero elements on the off diagonal terms arise from the assumed vehicular symmetry when viewed in the $x-y$ and $y-z$ planes. Because the vehicle lacks symmetry when viewed on the $x-z$ plane, this off diagonal term must be included as shown in figure 4.3. The second term represents the corrections from the parallel axis theorem and the fact that the actual body origin is not the body center of mass, but slightly behind and below as has been shown.

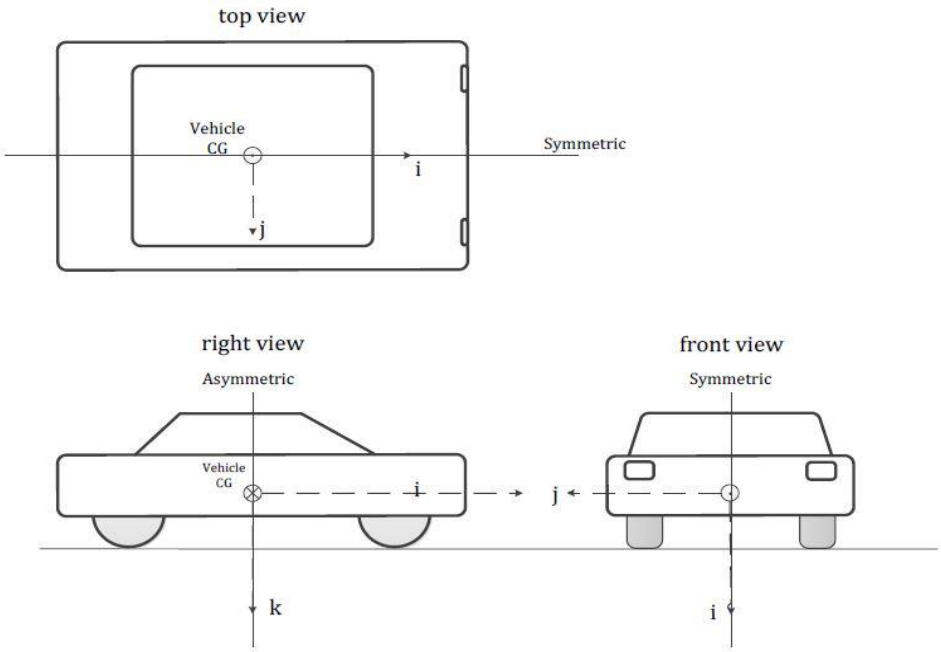

Figure 4.3: Symmetric and asymmetric planes of the vehicle [26].

Now that the moment of inertia tensor has been defined in the body frame of reference, it is necessary that it be projected into the chassis frame of reference. The rotation matrix for the transformation is given by:

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}. \tag{4.40}$$

The transformation itself is shown to be:

$$I_{sc} = RI_{sb}R^T. \tag{4.41}$$

Finally, recalling that the unsprung masses only rotate about the *z*-axis, their moment of inertia terms must be added to the moment of inertia tensor. These additional terms are shown to be:

$$I_{usc} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & I_{zzuf} + I_{zzur} + M_{uf}l_f^2 + M_{ur}l_r^2 \end{bmatrix}, \tag{4.42}$$

where the I terms are the moments of inertia about the unsprung mass centers and the terms that follow represent the parallel axis corrections. Therefore the final moment of inertia tensor is shown to be:

$$I_c = I_{sc} + I_{usc}. \tag{4.43}$$

Taking the time derivative of equation 4.38 and recalling that the pitch degree of freedom is neglected:

$$\frac{\partial H_c}{\partial t} = \frac{dH_c}{dt} + \Omega_c \times H_c, \tag{4.44}$$

$$\frac{\partial H_c}{\partial t} = (\dot{H}_x - H_y \Psi)i + (\dot{H}_z + H_y \Phi)k, \tag{4.45}$$

$$\begin{aligned}\frac{\partial H_c}{\partial t} = ((I_{xxs} + M_s h_s^2)\ddot{\Phi} &+ \left(I_{zxs} + M_s h_s l_{cgs}\right)\cos\Phi\ddot{\Psi} \\ &- \left(I_{zxs} + M_s h_s l_{cgs}\right)\sin\Phi\dot{\Psi}\dot{\Phi} \\ &- \left(I_{zzs} - I_{yys} - M_s h_s^2\right)\sin\Phi\cos\Phi\dot{\Psi}^2)i \\ &+ ((\left(I_{zxs} + M_s h_s l_{cgs}\right)\cos\Phi\ddot{\Psi} \\ &+ \left[\left(I_{yys} + M_s\left(h_s^2 + l_{cgs}^2\right)\right)\sin^2\Phi + \left(I_{zzs} + M_s l_{cgs}^2\right)\cos^2\Phi\right]\ddot{\Psi} \\ &+ \left(I_{zxs} + M_s h_s l_{cgs}\right)\sin\Phi\,\dot{\Phi}^2 + \left(I_{zzs} - I_{yys} - M_s h_s^2\right)k.\end{aligned} \tag{4.46}$$

Having defined the time derivative of the angular momentum, and recalling that:

$$\Sigma M = \frac{\partial H_c}{\partial t} = \frac{dH_c}{dt} + \Omega_c \times H_c, \tag{4.47}$$

the following equations of motion can be extracted. The first equation is the roll angular acceleration:

$$\begin{aligned}(I_{xxs} + M_s h_s^2)\ddot{\Phi} &= \left(\Sigma T_{xs} - \left(I_{xzs} - M_s h_s l_{cgs}\right)\cos\Phi\,\ddot{\Psi}\right. \\ &+ \left(I_{xzs} - M_s h_s l_{cgs}\right)\sin\Phi\dot{\Psi}\dot{\Phi} + (I_{zzs} - I_{yys} \\ &\left. - M_s h_s^2)\sin\Phi\cos\Phi\dot{\Psi}^2\right).\end{aligned} \tag{4.48}$$

The second equation is the yaw angular acceleration:

$$\begin{aligned}I_{zzo}\ddot{\Psi} = \left(\Sigma T_z - \left(I_{xzs} - M_s h_s l_{cgs}\right)\cos\Phi\,\ddot{\Psi}\right. \\ - \left(I_{xzs} - M_s h_s l_{cgs}\right)\sin\Phi\dot{\Phi}^2 \\ + \left(I_{zzs} - I_{yys} - M_s h_s^2\right)\sin\Phi\cos\Phi\dot{\Psi}^2 - M_s h_s a_x \sin\Phi \\ \left. - \left(I_{zzs} - I_{yys} - M_s h_s^2\right)\sin\Phi\cos\Phi\dot{\Psi}\dot{\Phi}\right).\end{aligned} \tag{4.49}$$

The sum of the torques acting on the sprung mass about the *x* axis is shown to be:

$$\Sigma T_{xs} = T_{\Phi f} + T_{\Phi r} + M_s g h_s \sin\Phi, \tag{4.50}$$

where

$$T_{\Phi f} + T_{\Phi r} = -(K_{\Phi f} + K_{\Phi r})\Phi - (B_{\Phi f} + B_{\Phi r})\dot{\Phi}. \tag{4.51}$$

Where $T_{\Phi f}$ and $T_{\Phi r}$ represent the net torque resulting from the suspension. The roll stiffness $K_{\Phi\star}$ and roll damping $B_{\Phi\star}$ are defined as:

$$K_{\Phi\star} = 0.766K_{s\star}\frac{t_f^2}{2} + K_{r\star}, \tag{4.52}$$

$$B_{\Phi\star} = 0.827B_{s\star}\frac{t_f^2}{2} + B_{r\star}. \tag{4.53}$$

Where $K_{s\star}$ is the suspension spring stiffness, $B_{s\star}$ is the shock damping coefficient, $K_{r\star}$ is the anti-roll bar stiffness, and $B_{r\star}$ is the anti-roll bar damping. The test vehicle is not mounted with anti-roll bars, so these values are zero. Recall that $\star$ denotes the front and rear of the vehicle.

The sum of torques about the *z* axis is given by:

$$\begin{aligned}\Sigma T_z = {} & \left(F_{ylf} + F_{yrf}\right)l_f - \left(F_{ylr} + F_{yrr}\right)l_r + \left(F_{xlf} + F_{xrf}\right)\frac{t_f}{2} + \left(F_{xlr} - F_{xrr}\right)\frac{t_r}{2} \\ & + M_{zlf} + M_{zrf} + M_{zlr} + M_{zrr},\end{aligned} \tag{4.54}$$

where the first four terms are the moments developed by the tire forces about the body and the last four terms are the tire self-aligning moments.

4.4 Dynamic Weight Transfer Forces

Having derived the equations of motion for both translation and rotation, it is obvious that the summation of forces and moments are necessary in order for each time step to be evaluated. These forces result from interactions of the tires with the ground, and are dependent upon the tire normal force. Because the chassis coordinate system is not an inertial (global) coordinate system, a handful of correction accelerations must be supplied before the coordinate system is valid. The normal acceleration terms used to simplify the longitudinal and lateral equations of motion are one set of accelerations that will develop dynamic weight transfer in the vehicle. The other accelerations are the tangential accelerations provided by driving or braking the tires. These accelerations serve to cancel whatever acceleration is being experienced by the frame of reference and therefore make it an inertial system and valid for the Newtonian laws of physics.

For any longitudinal correcting acceleration, the resulting normal force compensation is:

$$F_{zax} = \frac{M_s h_s a_{sx} + M_{uf} h_{uf} a_{ufx} + M_{ur} h_{ur} a_{urx}}{2L}. \tag{4.55}$$

These terms are understood as the moment balance shown in figure 4.4.

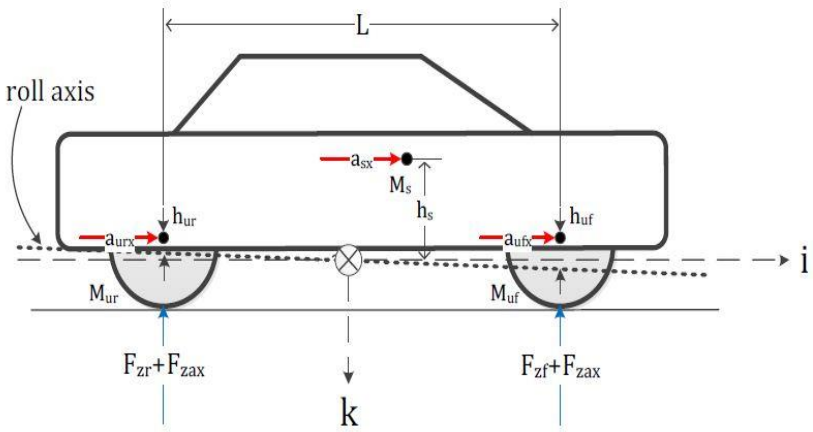

Figure 4.4: Longitudinal weight transfer force [26].

For lateral acceleration across the front of the vehicle, the normal force compensation is shown to be:

$$F_{zayf} = \frac{1}{t_f}\left(\frac{M_s l_r h_f}{2L} a_{sy} + M_{uf} h_{uf} a_{ufy}\right), \tag{4.56}$$

and the rear normal force compensation is shown to be:

$$F_{zayr} = \frac{1}{t_r}\left(\frac{M_s l_f h_r}{2L} a_{sy} + M_{ur} h_{ur} a_{ury}\right). \tag{4.57}$$

To compensate for the normal force due to roll it is shown that the necessary force is equal to:

$$F_{z\Phi f} = -\frac{1}{t_f}(K_{\Phi f}\Phi + B_{\Phi r}\dot{\Phi}), \tag{4.58}$$

$$F_{z\Phi f} = -\frac{1}{t_r}(K_{\Phi r}\Phi + B_{\Phi r}\dot{\Phi}), \tag{4.59}$$

for the rear half of the vehicle. The normal forces on each tire are then found by applying the equations 4.56 to 4.59 as follows:

$$F_{zlf} = \frac{Mgl_r}{2L} - F_{zax} + F_{zayf} + F_{z\Phi f}, \tag{4.60}$$

$$F_{zrf} = \frac{Mgl_r}{2L} - F_{zax} - F_{zayf} - F_{z\Phi f}, \tag{4.61}$$

$$F_{zlr} = \frac{Mgl_f}{2L} + F_{zax} + F_{zayf} + F_{z\Phi f}, \tag{4.62}$$

$$F_{zrr} = \frac{Mgl_f}{2L} + F_{zax} - F_{zayf} - F_{z\Phi f}. \tag{4.63}$$

With the normal force on each wheel determined, the longitudinal and lateral tire forces are derived. Several different methods exist that estimate the behavior of these forces under varying conditions. Two tire models have been described in literature review section of this report.

Chapter 5

System Identification

This chapter focuses on system identification of the scaled test vehicle that was used in this study. Mainly the focus was to identify parameters of the vehicle model derived in the previous chapter (chapter 4), so that the model could fit the test vehicle platform used in this study.

5.1 Experimental Apparatus

5.1.1 Scaled Test Vehicle

In order to properly conduct scaled vehicle testing it is necessary to find a vehicle system that can approximate the operation of a full sized vehicle. The vehicle should be properly scaled in dimension, re-configurable, incorporate a suspension/chassis system of the same architecture and have a powerful drive system that would allow it to perform aggressive testing manoeuvres. Due to cost and time constraints to find/build such an ideal scaled vehicle, an off the shelf vehicle was selected which would provide good results in accordance with the main aim of this research, i.e. rollover prevention and path following. The selected test vehicle was the River Hobby VRX Sword Brushless vehicle ($1/10^{th}$ scaled Remote Controlled (RC) car), which is shown in figure 5.1.

The selected River Hobby VRX Sword Brushless vehicle can reach speeds in excess of 25 km/h and is driven by a single brushless DC motor. The vehicle is of four wheel drive system and employs full braking on all wheels. It incorporates an Ackerman steering arrangement as well as a fully independent double wishbone suspension system that utilizes hydraulic dampers with coaxial springs to control body motions. The steering system is actuated by a servo motor and the steering angle maximum/minimum value is ± 17 degrees. Overall, the vehicle system closely resembles that of a full size vehicle and therefore made the River Hobby Sword an ideal platform for this study.

Figure 5.1: River Hobby VRX Sword 1/10th scaled vehicle.

5.1.2 MPU-6050

To measure outputs of the scaled vehicle described, an Inertia Measurement Unit (IMU) was needed. For this study, an IMU called MPU-6050 (figure 5.2) was selected because of its great specifications coupled with its low cost. The MPU-6050 is an integrated 6-axis motion tracking device that combines a 3-axis gyroscope, 3-axis accelerometer, and a Digital Motion Processor (DMP) all in a small 4x4x0.9 mm package. With its dedicated I^2C sensor bus, it can directly accept inputs from an external 3-axis compass to provide a complete 9-axis motion fusion output if need arises.

The MPU-6050 features three 16-bit analog-to-digital converters (ADCs) for digitizing the gyroscope outputs and three 16-bit ADCs for digitizing the accelerometer outputs. For precision tracking of both fast and slow motions, the parts feature a user-programmable gyroscope full-scale range of ±250, ±500, ±1000, and ±2000°/sec (degrees per second) and a user-programmable accelerometer full-scale range of ±2g, ±4g, ±8g, and ±16g. An on-chip 1024 Byte FIFO buffer helps lower system power consumption by allowing the system processor to read the sensor data in bursts and then enter a low-power mode as the MPU collects more data. With all the necessary on-chip processing and sensor components required to support many motion-based use cases, the MPU-6050 uniquely enables low-power motion interface applications in portable applications with reduced processing requirements for the system processor. By providing an integrated motion fusion output, the DMP in the MPU-6050 offloads the intensive motion processing computation requirements from the system processor, minimizing the need for frequent polling of the motion sensor output. Communication with all registers of the device is performed using I^2C at 400 kHz.

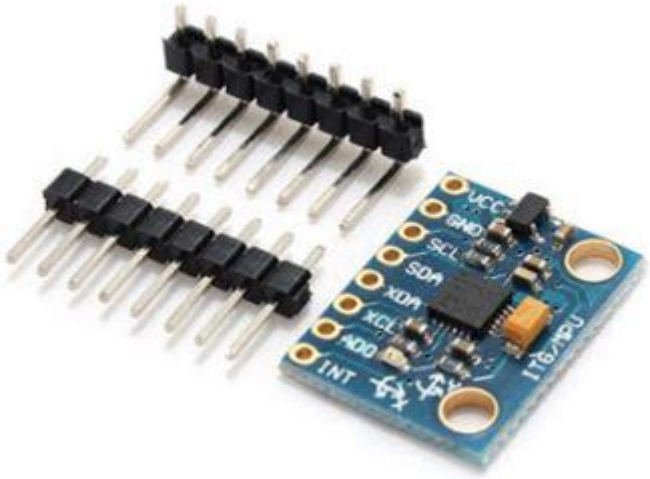

Figure 5.2: 6 DOF MPU-6050 with connecting pins.

5.1.3 Arduino UNO

To facilitate data acquisition from the MPU-6050, a microcontroller known as Arduino Uno (figure 5.3) was used in conjunction with an external Secure Data card (SD card) to store the data. The driving motor voltage and steering signal were captured directly by the Arduino by tapping the signal wires and using the Arduino “read” pins. The selection of this controller was based on its ability to achieve the intended use as well as cost. Many microcontrollers, including the Arduino Uno, would achieve the task, but this was the cheapest. More information about the Arduino Uno can be found on the following web link: https://store.arduino.cc/arduino-uno-rev3.

Figure 5.3: Arduino Uno Board.

5.1.4 Complete Test Platform

The MPU-6050 IMU and SD card were connected to the Arduino Uno board. Then the whole setup was mounted on the scaled vehicle model as shown in figure 5.4.

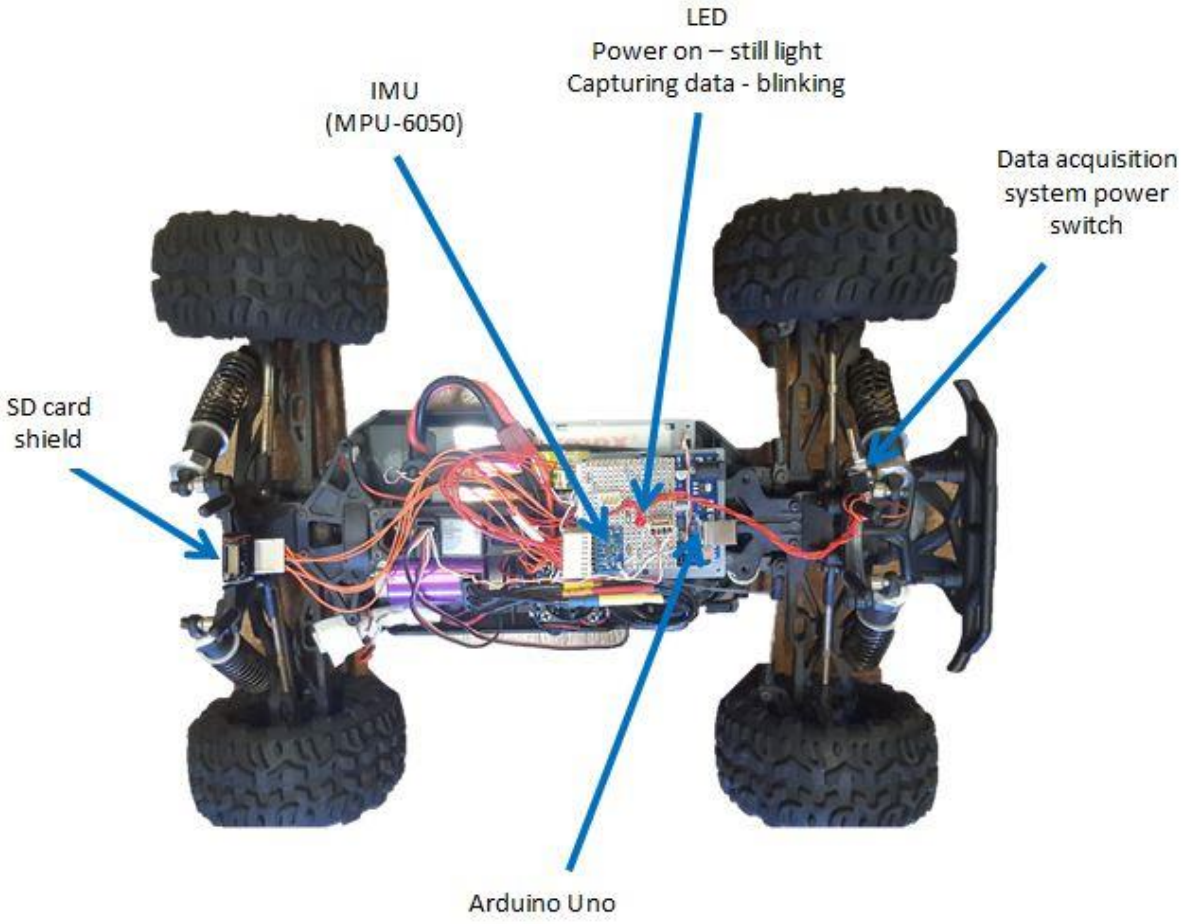

Figure 5.4: Showing complete test platform.

The MPU-6050 IMU was mounted at the vehicle's Centre of Gravity (CoG) which eliminates the necessity for any manipulation or transformation of the data. The calculation of the CoG is shown later in this chapter of the report. The Arduino was run directly from the standard 7.4V LiPo battery, which was also used to power the vehicle driving motor and steering servo motor. The MPU-6050 IMU and the SD card were powered from the Arduino Uno using their standard connection circuits.

5.2 Experiment Design and Results

In order to capture the lateral and longitudinal states of the physical scaled vehicle platform, experiments were performed. The experiments were designed in such a way that both the longitudinal and lateral dynamics of the vehicle are excited during experiments.

The inputs to the test platform were the voltage to the driving motor and steering command to the steering servo. The outputs which were measured are longitudinal

acceleration, lateral acceleration, roll angle and yaw angle. Longitudinal velocity, which is also of interest in this study, was calculated by integration of longitudinal acceleration.

In order to excite vehicle lateral dynamics, a number of standardized steer manoeuvres have been developed. Among the manoeuvres, simple step steer (SSS) and double lane change (DLC) manoeuvres are commonly used techniques to evaluate vehicle handling and performance in research and in industry [26]. These step manoeuvres are also mostly used to test the designed controller performance in research, of which examples can be found in [7], [20], [21]. Similarly, in this study simple step steer and double lane change manoeuvres were used during system identification of the platform vehicle.

To have sufficient data for better analysis and reduce measurement errors, the experiments were done at two different velocities, which are 3 m/s and 3.8 m/s, and two different steering angles which are -10 degrees and ± 17 degrees (simple step steer and double lane change manoeuvre respectively). The maximum/minimum value for steering angle was ± 17 degrees; they were physically measured on the platform. The velocity inputs were reached at by doing a number of experiments and determining at least two different high safe values that provided sufficient information of the needed data. Where high safe values mean those values just before rollover of the platform happens. The combination of these two inputs during experiments was as shown in table 5.1.

Table 5.1: System identification input values.

Experiment number	Input velocity (m/s)	Input steering angle (degrees)
Experiment-1 (DLC)	3	± 17
Experiment-2 (SSS)	3.8	-10

The following are the results of the experiments. The platform velocity and steering angles are shown alongside their associated roll angles. For the outputs, only roll angle graphs are shown for clarity and brevity purposes.

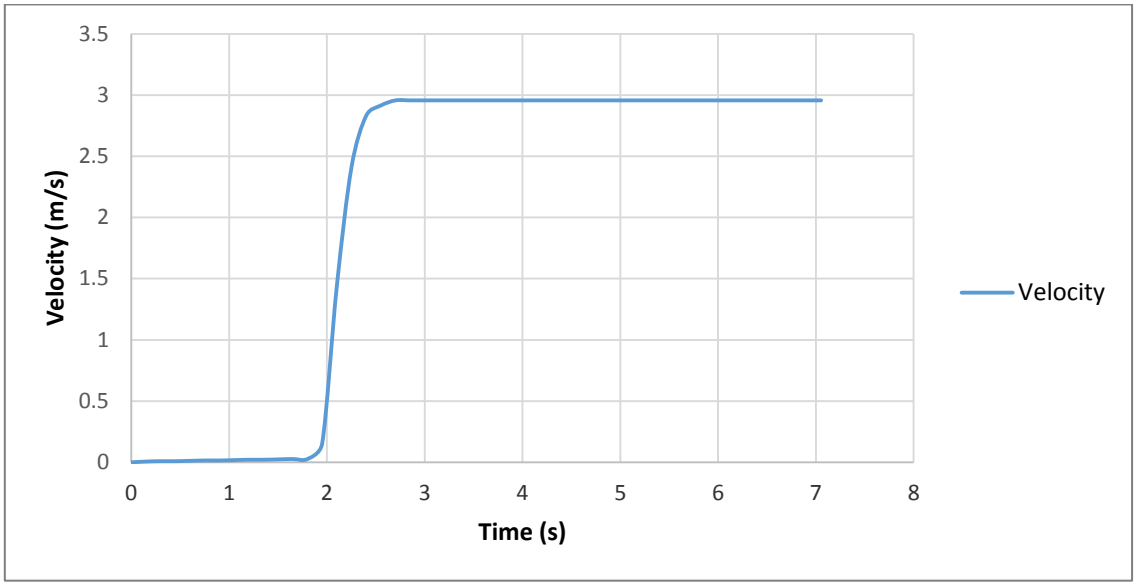

Figure 5.5: Showing experiment-1(DLC) longitudinal velocity input rising from 0 to close to 3 m/s and remains there.

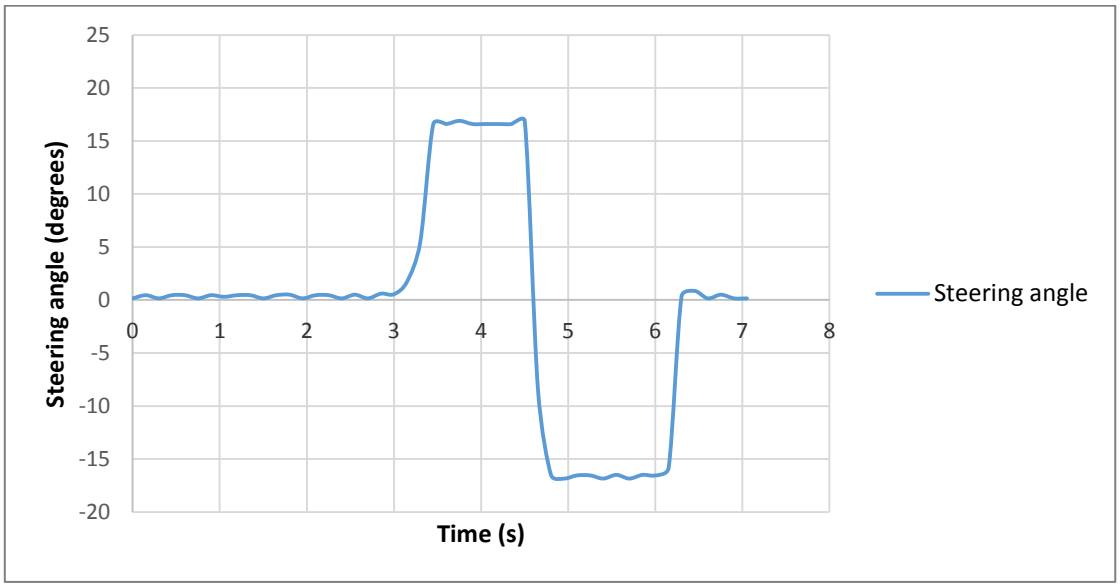

Figure 5.6: Showing experiment-1 (DLC) Steering angle input; shown is a positive step (17 degrees) followed by negative step (-17 degrees) in steering angle.

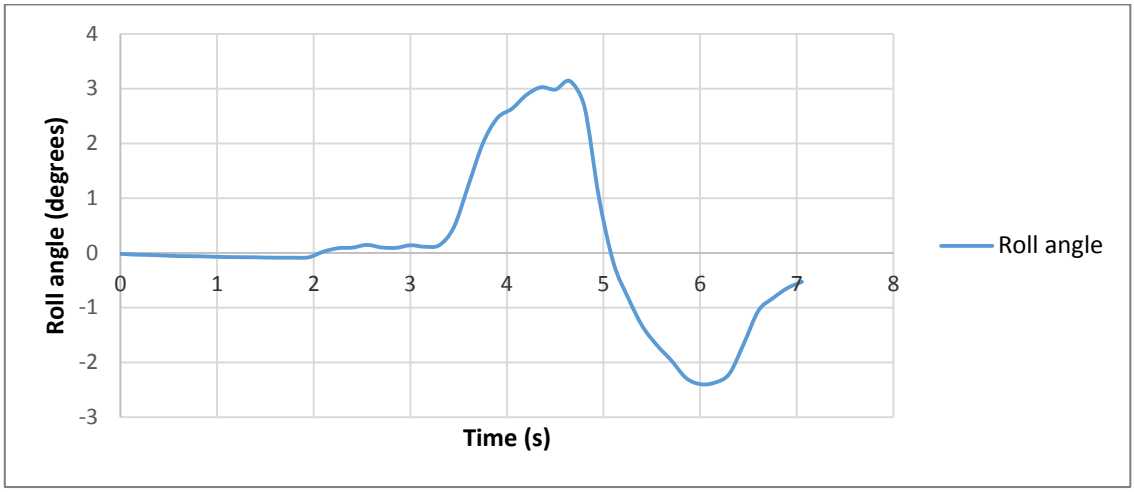

Figure 5.7: Showing experiment-1 (DLC) roll angle output; shown is a positive roll angle followed by a negative roll angle.

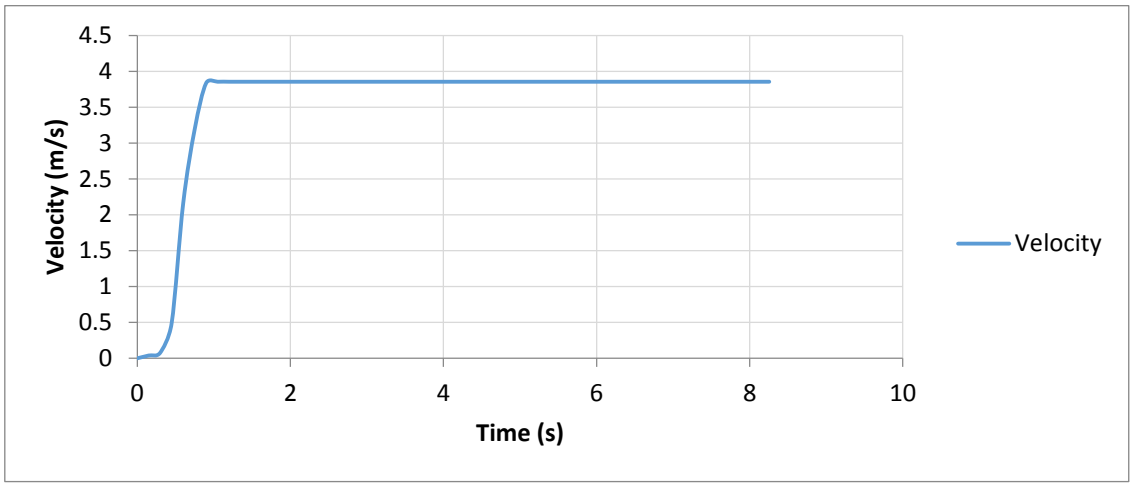

Figure 5.8: Showing experiment-2 (SSS) longitudinal velocity input rising from 0 to 3.8 m/s and remains there.

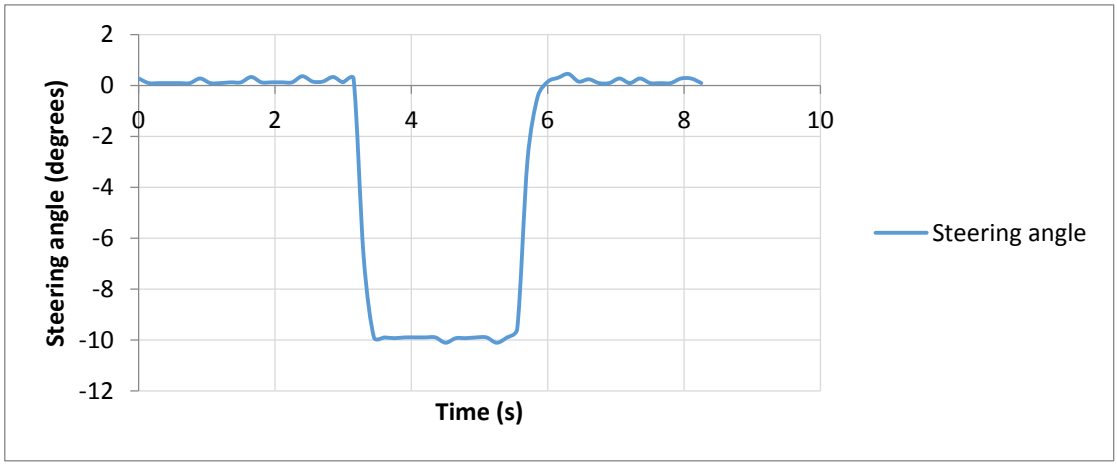

Figure 5.9: Showing experiment-2 (SSS) steering angle input; shown is a negative step (-10 degrees).

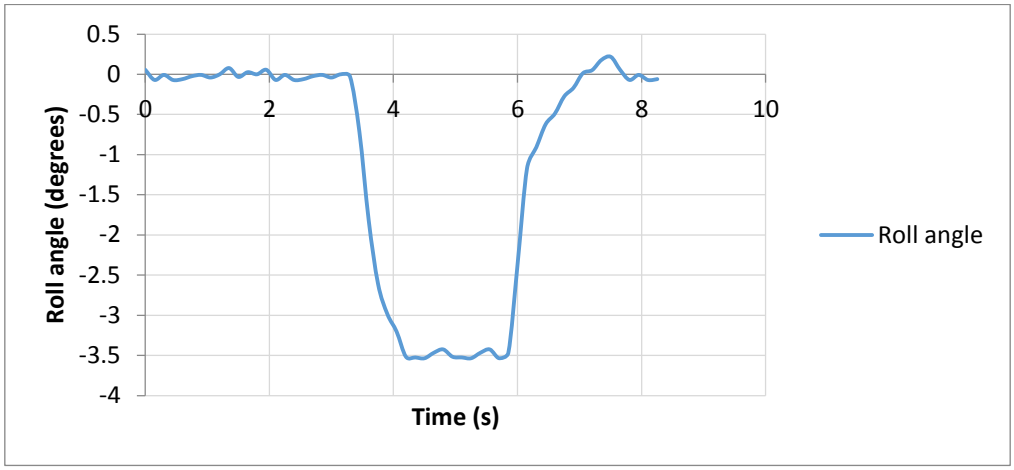

Figure 5.10: Showing experiment-2 (SSS) roll angle output; shown a negative roll angle output.

The vehicle was first driven in the forward direction up to the designed experimental velocity, and then the steering angle input was applied at that velocity. This can be seen in the figures 5.5 and 5.6 similarly in figures 5.8 and 5.9.

5.3 Nonlinear Least Squares (Scilab)

After the experiment in the previous section, the next step was to identify the parameters of the model and this section is devoted to that.

An open source software known as Scilab was used in this study to calculate the model parameters. Scilab implements a number of optimisation algorithms discussed in literature review chapter under system identification and parameter estimation section. Examples of implemented algorithms are *gradient decent, quasi-Newton* and *Neldermead* [66]. In addition to these, *genetic algorithms* and *simulated annealing* are also implemented.

In this study nonlinear least square was used to calculate the parameters. The nonlinear least squares in Scilab implements *quasi-newton* and *gradient decent* algorithms for optimisation. Both of these were used and results were fairly the same. The final results used in this study were generated by the *quasi-newton* algorithm.

The least square implementation solves problems of the following form:

Suppose that $y = f(x)$ is a function from the space of p-dimensional vectors x to that of q-dimensional vectors y. Then if x is calculated as

$$\min_x \|f(x)\|^2 = f(x)'f(x) = \sum_{i=1}^{p} f_i(x)^2, \tag{5.1}$$

such a solution is called a least squares solution. Notice that this allows for $p > q$. That is, there are more equations than there are unknowns.

To use the least square functionality of Scilab, the model derived in the previous chapter was supposed to be rearranged by putting it in nonlinear state space form. After putting in state space, the model was simulated by applying similar inputs (driving motor voltage and steering angle) which were applied during data collection in the physical experiments. The difference between the output from the simulated model and the actual data from experiment was represented by the function $f(x)$ in equation 5.1.

Nonlinear State Space Model

The vehicle model derived in chapter 4 was rewritten to match the vehicle model which was presented by Taehyun Shim and Chinar Ghike in [39]. The model assumes small angle approximation for sin and cosine functions, which reduces computational time, but it is still nonlinear. It was validated against the 14 degrees of freedom nonlinear vehicle model, the results matched fairly well as presented in [39]. The model used in this study was then compactly written as:

$$\dot{\xi}(t) = f\big(\xi(t), u(t)\big). \tag{5.2}$$

Where $\xi(t) \in R^n$ is representing the state of the vehicle and $u(t) \in R^m$ is representing the input to the vehicle. The number of states is given by $n = 12$ and the number of inputs is given by $m = 2$. The twelve states of the vehicle are as follows: lateral and longitudinal velocities given in the body frame, yaw angle, yaw rate, roll angle, roll rate, lateral and longitudinal vehicle coordinates in the inertial frame and the angular velocity on the four wheels. These states are denoted as $\xi = [\dot{y}, \dot{x}, \Psi, \dot{\Psi}, \Phi, \dot{\Phi}, Y, X, \omega_{f,l}, \omega_{f,r}, \omega_{r,l}, \omega_{r,r}]'$ respectively. Where $\omega_{f,l}, \omega_{f,r}, \omega_{r,l}\ and\ \omega_{r,r}$ denotes the angular velocity on the front left, front right, rear left and rear right wheels, respectively. For ease of notation, the front or rear axle is denoted by the subscript $\star \in \{f, r\}$ and the left or right side of the vehicle is denoted by $\bullet \in \{l, r\}$ in this report. The two inputs to the vehicle system are $u = [\delta_f, T]'$ where δ_f is the front steering angle and T is the total braking or tractive torque on the four wheels. Positive T denotes driving torque while negative T denotes braking torque.

In particular, the lateral and longitudinal tire forces in tire frame are therefore represented as $F_{c\star,\bullet}$ and $F_{l\star,\bullet}$ respectively. The components of the tire forces along the lateral and longitudinal vehicle axes are represented as $F_{y\star,\bullet}$ and $F_{x\star,\bullet}$ respectively. The wheel slip angles, which are defined later in this section, are represented as $a_{\star,\bullet}$ .The distances from the CoG to the front and rear axles are represented by l_f and l_r respectively. The distance from the CoG to the left/right side at the front and rear wheels is represented by $\frac{t_f}{2}$ and $\frac{t_r}{2}$ respectively. The state equations are as follows:

$$M\dot{U}_{ox} = \Sigma F_x + \big(M_{uf} l_f - M_{ur} l_r\big)\dot{\psi}^2 - 2h_{rc} M\,\dot{\psi}\dot{\phi} + M\,\dot{\psi} U_{oy}, \tag{5.3a}$$

$$M\dot{U}_{oy} = \Sigma F_y + \left(M_{ur}l_r - M_{uf}l_f\right)\ddot{\psi} + h_{rc}M\ddot{\phi} - M\dot{\psi}U_{oy} - M\dot{\psi}U_{ox}, \tag{5.3b}$$

$$I_{zz}\ddot{\psi} = \left(F_{ylf} + F_{yrf}\right)l_f - \left(F_{ylr} + F_{yrr}\right)l_r + \left(F_{xlf} + F_{xrf}\right)\frac{t_f}{2} + \left(F_{xlr} - F_{xrr}\right)\frac{t_r}{2} + \left(M_{ur}l_r - M_{uf}l_f\right)\left(\dot{U}_{oy} + \dot{\psi}U_{ox}\right) - I_{xz}\ddot{\phi}, \tag{5.3c}$$

$$\begin{aligned}\left(I_{xx} + Mh_{rc}^2\right)\ddot{\phi} + I_{xz}\ddot{\psi} &= Mgh_{rc}\phi - \left(k_{\phi f} + k_{\phi r}\right)\phi - \left(b_{\phi f} + b_{\phi r}\right)\dot{\phi} \\ &\quad + h_{rc}M\left(\dot{U}_{oy} + \dot{\psi}U_{ox}\right),\end{aligned} \tag{5.3d}$$

$$\dot{Y} = \dot{x}\sin\Psi + \dot{y}\cos\Psi, \tag{5.3e}$$

$$\dot{X} = \dot{x}\cos\Psi - \dot{y}\sin\Psi, \tag{5.3f}$$

$$I_w\dot{\omega}_{f,l} = -F_{lf,l}r_w + T_{f,l} - b_w\dot{\omega}_{f,l}, \tag{5.3g}$$

$$I_w\dot{\omega}_{f,r} = -F_{lf,r}r_w + T_{f,r} - b_w\dot{\omega}_{f,r}, \tag{5.3h}$$

$$I_w\dot{\omega}_{r,l} = -F_{lr,l}r_w + T_{r,l} - b_w\dot{\omega}_{r,l}, \tag{5.3i}$$

$$I_w\dot{\omega}_{r,r} = -F_{lr,r}r_w + T_{r,r} - b_w\dot{\omega}_{r,r}. \tag{5.3j}$$

Where

$$\Sigma F_x = F_{xlf} + F_{xrf} + F_{xlr} + F_{xrr}, \tag{5.4a}$$

$$\Sigma F_y = F_{ylf} + F_{yrf} + F_{ylr} + F_{yrr}, \tag{5.4b}$$

$$h_{rc} = \frac{h_{rcf}l_r + h_{rcr}l_f}{l_f + l_r}. \tag{5.4c}$$

The normal forces at the four tires are determined as in chapter 4 of this report as follows:

$$F_{zlf} = \frac{Mgl_r}{2L} - F_{zax} + F_{zayf} + F_{z\Phi f}, \tag{5.5a}$$

$$F_{zrf} = \frac{Mgl_r}{2L} - F_{zax} - F_{zayf} - F_{z\Phi f}, \tag{5.5b}$$

$$F_{zlr} = \frac{Mgl_f}{2L} + F_{zax} + F_{zayf} + F_{z\Phi f}, \tag{5.5c}$$

$$F_{zrr} = \frac{Mgl_f}{2L} + F_{zax} - F_{zayf} - F_{z\Phi f}. \tag{5.5d}$$

Where the constant M is the vehicle's total mass, L is distance between front and rear axle, $M_{ur}, M_{uf}, h_{rc}, h_{rcr}, and\ h_{rcf}$ are rear unsprung mass, front unsprung mass, vehicle roll centre, rear roll centre and front roll centre heights respectively. I_{xx}, I_{zz}, I_{xz} are the vehicle's rotational inertia about the x-axis, z-axis and their product, respectively. I_w includes the wheel and driveline rotational inertias. r_w is the radius of the wheel. b_w is the wheel damping coefficient. The x and y components of tire forces, $F_{x\star,\bullet}$ and $F_{y\star,\bullet}$, are computed as follows:

$$F_{y\star,\bullet} = F_{l\star,\bullet} \sin\delta_\star + F_{c\star,\bullet} \cos\delta_{\star,} \tag{5.6a}$$

$$F_{x\star,\bullet} = F_{l\star,\bullet} \cos\delta_\star - F_{c\star,\bullet} \sin\delta_\star . \tag{5.6b}$$

For the vehicle platform used in this study, only the steering angle at the front wheels can be controlled. Moreover, the front left and front right wheel steering angles are assumed to be the same. i.e., $\delta_{f,l} = \delta_{f,r} = \delta_f$ and $\delta_{r,\bullet} = 0$.

The lateral and longitudinal tire forces $F_{l\star,\bullet}$ and $F_{c\star,\bullet}$ are given by

$$F_{c\star,\bullet} = f_c(\alpha_{\star,\bullet}, s_{\star,\bullet}, \mu, F_{z\star,\bullet}), \tag{5.7a}$$

$$F_{l\star,\bullet} = f_l(\alpha_{\star,\bullet}, s_{\star,\bullet}, \mu, F_{z\star,\bullet}), \tag{5.7b}$$

where α is the slip angle of the tire, s is the slip ratio, μ is the friction coefficient and F_z is the normal force. The Pacejka tire model is used to model F_c and F_l in equation 5.7 at the four tires. This complex, semi-empirical model is able to describe the tire behavior over the linear and nonlinear operating ranges of slip ratio, tire slip angle and friction coefficient. The tire slip angle $\alpha_{\star,\bullet}$ in equation 5.7 denotes the angle between the tire velocity and its longitudinal direction. It can be expressed as:

$$\alpha_{\star,\bullet} = \arctan\left(\frac{u_{c\star,\bullet}}{u_{l\star,\bullet}}\right), \tag{5.8}$$

where $u_{c\star,\bullet}$ and $u_{l\star,\bullet}$ are the lateral and longitudinal wheel velocities computed from:

$$u_{c\star,\bullet} = u_{y\star,\bullet} \cos\delta_\star - u_{x\star,\bullet} \sin\delta_\star , \tag{5.9a}$$

$$u_{l\star,\bullet} = u_{y\star,\bullet} \sin\delta_\star + u_{x\star,\bullet} \cos\delta_\star , \tag{5.9b}$$

$$u_{yf,\bullet} = \dot{y} + a\dot{\Psi} \qquad u_{x\star,l} = \dot{x} - c\dot{\Psi}, \tag{5.9c}$$

$$u_{yr,\bullet} = \dot{y} - b\dot{\Psi} \qquad u_{x\star,r} = \dot{x} + c\dot{\Psi}. \tag{5.9d}$$

The slip ratio is defined as:

$$s_{\star,\bullet} = \begin{cases} \dfrac{r_w \omega_{\star,\bullet}}{u_{l\star,\bullet}} - 1 & \text{if } u_{l\star,\bullet} > r_w \omega_{\star,\bullet}, \quad u \neq 0 \text{ for braking} \\ 1 - \dfrac{u_{l\star,\bullet}}{r_w \omega_{\star,\bullet}} & \text{if } u_{l\star,\bullet} < r_w \omega_{\star,\bullet}, \quad \omega \neq 0 \text{ for driving} \end{cases} \tag{5.10}$$

Recalling the Magic Formula from literature review section, which is given by

$$y = D \sin(C \arctan\{Bx - E(Bx - \arctan Bx)\}) \tag{5.11}$$

Where y represents the output force either F_c or F_l.

In figure 5.11 is an example showing normalized relationship between cornering force and slip angle for specific values of B, C and E for the Magic formula in equation 5.11.

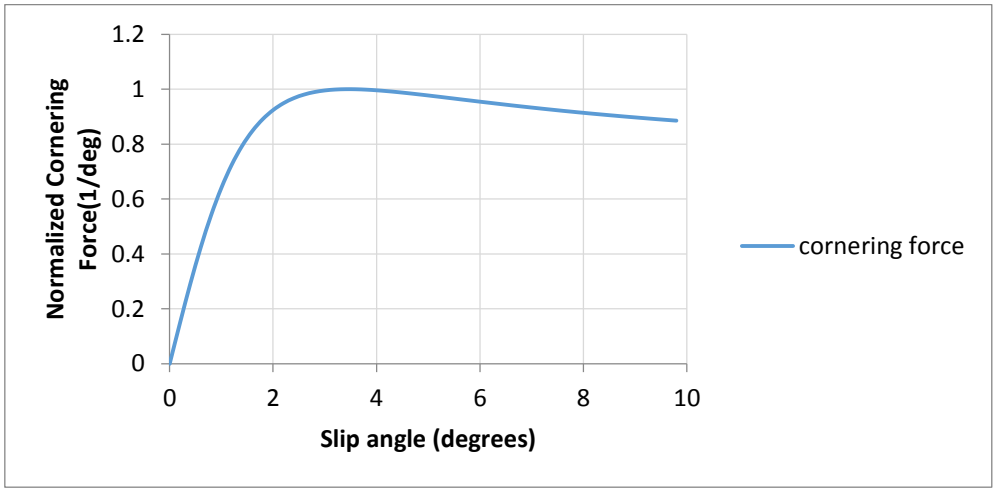

Figure 5.11: Example of Cornering force curve of the Magic Formula (B=0.5, C=2.5, D=1, E=1).

It is noted that for small slip angles, the cornering force increases almost linearly with the slip angle. Therefore, for small angles, the approximation $\sin\theta \cong 0$ and $\arctan\theta \cong \theta$ was used in this study. The aim was to reduce computation time. It was assumed the vehicle would not reach the nonlinear region of the curve (i.e. tire saturation). Using this approximation, the Magic Formula is reduced to:

$$F_c = DCB\alpha = C_\alpha \alpha. \tag{5.12}$$

This linear approximation is the simplest tire model. Under this model, the lateral and longitudinal tire forces are considered to vary linearly with slip angle and slip respectively.

$$\begin{aligned} F_{c\star,\bullet} &= C_{\alpha\star,\bullet}\alpha_{\star,\bullet}, \\ F_{l\star,\bullet} &= l_{s\star,\bullet}s_{\star,\bullet}. \end{aligned} \tag{5.13}$$

The friction coefficient μ is assumed to be a known constant and is the same at all four wheels.

Finally, after putting the vehicle model in a form that the least squares function accepts, the remaining task was to determine initial values of the parameters. This is important because the algorithms used within least squares function, usually converge to values that are close to the initial values supplied to the function. This has been explained in the literature review section of this report.

Therefore, some parameters of the model were determined by performing experiments while others, which were not so straightforward to calculate experimentally, were estimated and supplied to the least squares solver. The parameters like front and rear track width, tire radii, distance of front and rear axles from centre of gravity (CoG) z-axis, were measured using a ruler. Total mass of the test platform vehicle was measured using a scale. Position of centre of gravity was necessary to calculate as well. This is the point at which the Inertia Measurement Unit (IMU) was placed as explained in earlier sections of this chapter.

Calculating the Center of Gravity (CoG) location

Calculating the CoG was done experimentally as it is shown in the figures 5.12 and 5.13. It was assumed that the vehicle is symmetric about a longitudinal plane of symmetry, and the test platform vehicle was weighted on a scale in two different positions as shown in the figures 5.12 and 5.13. The axle loads denoted by Wf_1 and Wr_1 were weighted with the vehicle on a horizontal surface. Similarly, Wf_2 and Wr_2 were measured when one axle was raised by amount of"a". There are different recommended values of "a" depending on vehicle type as described in [67], [68]. W, WB_1, WB_2 represent vehicle weight, wheel base in first experiment and wheel base in second experiment respectively. The difference of the wheel bases measurements are illustrated in the figures 5.12 and 5.13. Since the vehicle was in static position, the summation of all moments about a point must be zero. Therefore, the center of gravity distance from rear axle (referring to figure 5.12) was calculated as follows:

$$xr_1 = \frac{Wf_1 \times WB_1}{Wf_1 + Wr_1}. \tag{5.14}$$

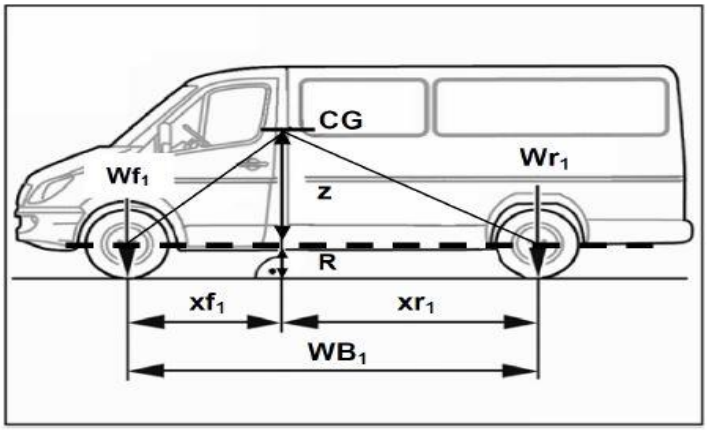

Figure 5.12: Illustrating the calculation of wheel axle distances from the CoG vertical axis [67].

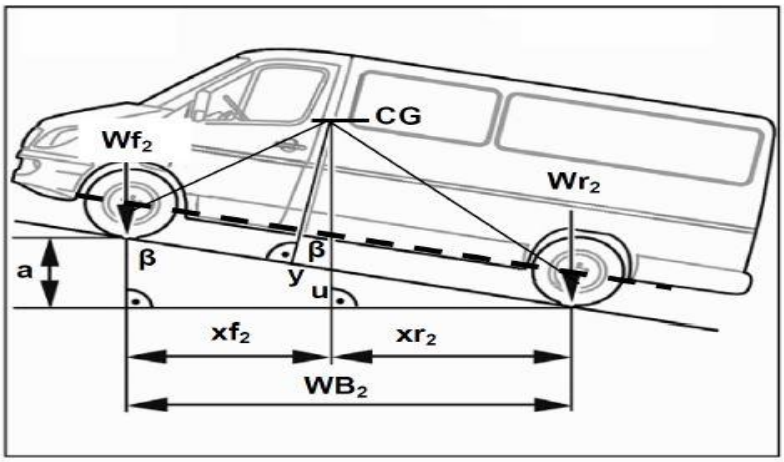

Figure 5.13: Illustrating the calculation of CoG height [67].

For raised axle test (figure 5.13), the center of gravity height was obtained by using the following equation:

$$h_{CoG} = r_w + \tan\left(\cos^{-1}\left(\frac{a}{WB_1}\right)\right) \times \left(\frac{Wf_1 \times WB_1}{W} - \frac{Wf_2 \times WB_1}{W}\right). \quad (5.15)$$

In which r_w is static radius height of front and rear wheels.

Model Validation

After the nonlinear least square function was invoked, the values of the model parameters were determined. The parameters and their values are in appendix B of this report.

To check the closeness of the model to the experimental platform, another experiment was conducted. In this experiment, different set of inputs, rather than those used to calculate the model parameters, were used. This was meant to check whether the model can fit the data which is different from the data it was derived from. The model was simulated using Scilab. During simulation, the inputs to the model were designed in such a way that they at least match the inputs that were actually applied in real experiment with the platform. The simulation and experimental results have been plotted on the same graphs for easy observation and analysis. The experiment was done at a velocity of 3.5 m/s and a step of 12 degrees steering angle. The following are the plots of results showing longitudinal velocity, steering input and roll angle.

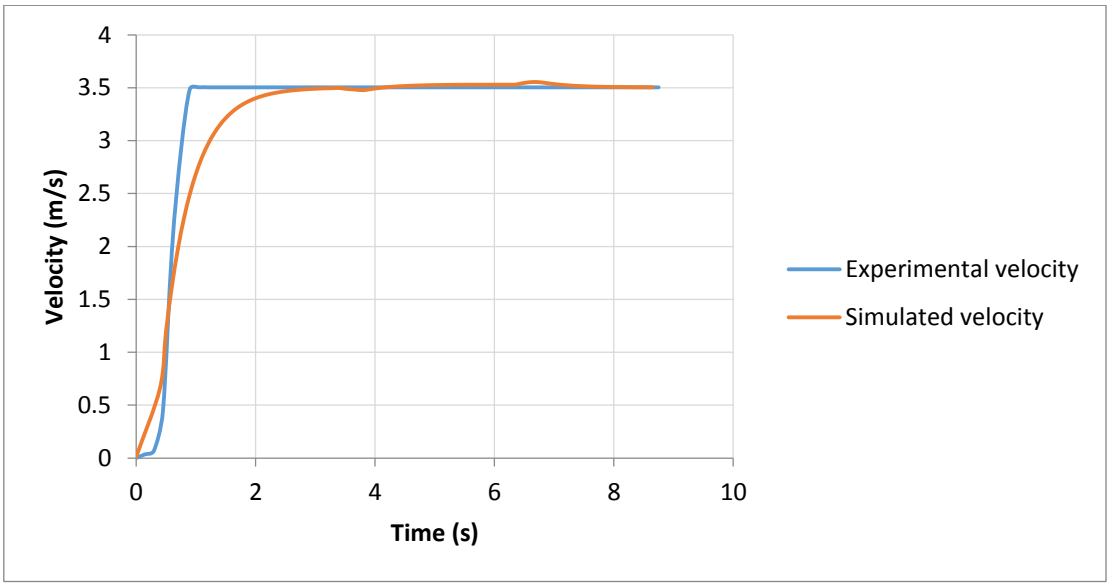

Figure 5.14: Showing experimental velocity (average 3.5 m/s) and simulated velocity (average 3.5 m/s) input plotted against time on same graph.

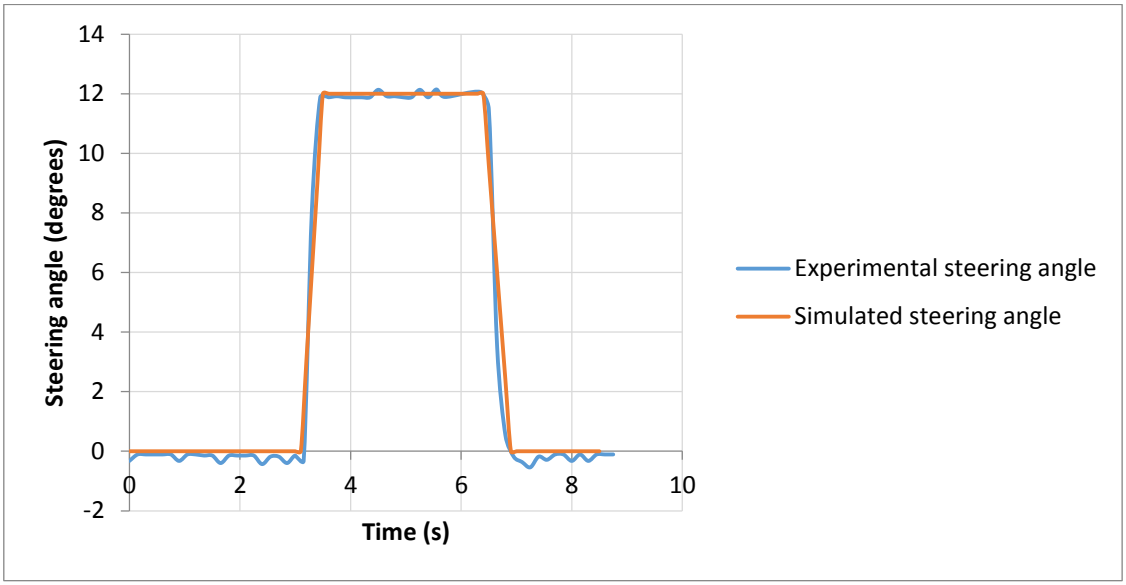

Figure 5.15: Showing experimental step steering angle (12 degrees) and simulated steering angle (12 degrees) inputs plotted against time on same graph.

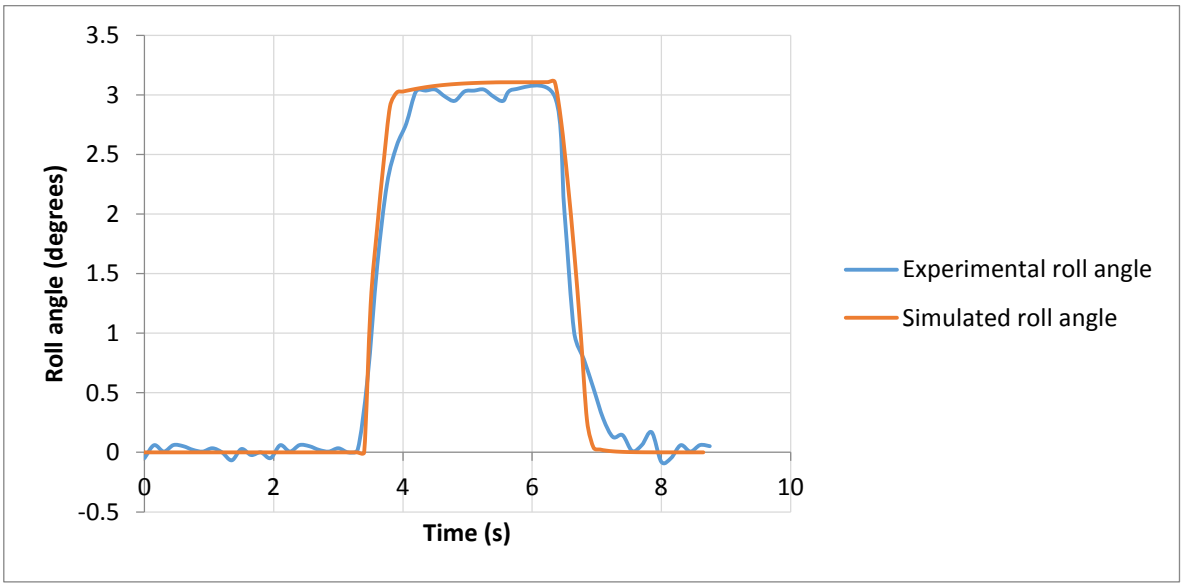

Figure 5.16: Showing experimental roll angle and simulated roll angle outputs plotted against time on same graph.

From the graphs shown, it was concluded that the mathematical model closely matches the physical platform behavior (valid model). The small discrepancies are due to noise in the experimental environment.

Chapter 6

Vehicle Stability Limit

In this chapter, the stability of the scaled vehicle model is analysed. Stability analysis is done after the model is identified and validated as done in the previous chapter which focused on vehicle system identification. Stability limit identification is necessary because the controller needs to be given these limits so that it is able to prevent the vehicle from going into unstable region. In this study, the stability limit values were given to the controller to avoid the vehicle from rolling over, and also to make sure the vehicle followed a defined reference trajectory without spinning out (yaw rate stability).

6.1 Roll and Yaw Stability

As highlighted in the literature review chapter of this report, there are a number of approaches available in literature to detect roll instability (onset of rollover). Some of these are roll angle measurement, load-transfer based methods, energy-based methods and lateral acceleration-based methods. In this study, roll angle measurement method was used. The Inertia Measurement Unit (IMU) described previously in system identification chapter is capable of measuring the roll angle directly.

To determine stability limit through simulation of the model, Load Transfer Ratio (LTR) was used. The idea of LTR has been explained in this report in the literature review chapter; the formula is captured here again for continuity and completeness.

The load transfer ratio (LTR) is defined as the difference between the normal forces on the right and left hand sides of the vehicle divided by their sum:

$$LTR = \frac{F_{zR} - F_{zL}}{F_{zR} + F_{zL}}. \tag{6.1}$$

An LTR value of ± 1 symbolises Total Wheel Lift (TWL) which determines the onset of a rollover event. In practice usually a value between ± 0.5 and ± 0.8 is used for control purposes.

As a way of finding the best value of LTR which could give meaningful results during controller testing, as well as achieving the goal of preventing the vehicle from roll over, a number of simulations of the model were done. These simulations were designed and performed by taking into consideration the limits of the physical vehicle platform. More insight is given after simulation results in this chapter.

When it comes to vehicle stability, lateral dynamics are of more important as compared to longitudinal dynamics (vertical dynamics are assumed to be at minimum in this study as explained in previous chapters of this report). The lateral dynamics stability is usually a function of mainly the forward (longitudinal) velocity and steering angle. Therefore, although the actual input to physical platform is voltage (motor voltage), in this analysis velocity instead, is treated as input in addition to steering angle. This change in reference input does not by any means change the results. This is because the transfer characteristic between voltage and torque (which results into platform velocity) of a DC motor is linear.

In reference to what has been said in the previous paragraphs, three simulations were done at two different input velocities and two different input steering angles. The combinations of the two inputs (velocity and Steering angle) are summarized in table 6.1.

Table 6.1: Simulations and their associated velocity and steering inputs.

Simulation	**Velocity (m/s)**	**Step steering angle (degrees)**
Simulation 1	4.6	0 ⟶ 5 ⟶ -5 ⟶ 0
Simulation 2	9.2	0 ⟶ 5 ⟶ -5 ⟶ 0
Simulation 3	9.2	0 ⟶ 10 ⟶ -10 ⟶ 0

Simulations

The following are the results of the simulations in the order they appear in table 6.1. The following abbreviations are used and there meanings are as follows:
$Fzlf$: Normal force at the left front tire.
$Fzrf$: Normal force at the right front tire.
$Fzlr$: Normal force at the left rear tire.
$Fzrr$: Normal force at the right rear tire.

Graphical results appear in the following order: velocity and steering inputs are plotted first, then tire normal forces, and finally the roll angle and yaw rate are plotted. For the sake of brevity and clarity, the two inputs (velocity and steering angle) have been plotted on the same graph for each simulation. Similarly the roll angle and yaw rate. It makes interpretation easy when tire forces, for example for left and right rear tires, are plotted on the same graph; this has also been done in the results shown.

- **Simulation 1**
  (Velocity of 4.5 m/s and step steering angles of +5 and -5 degrees)

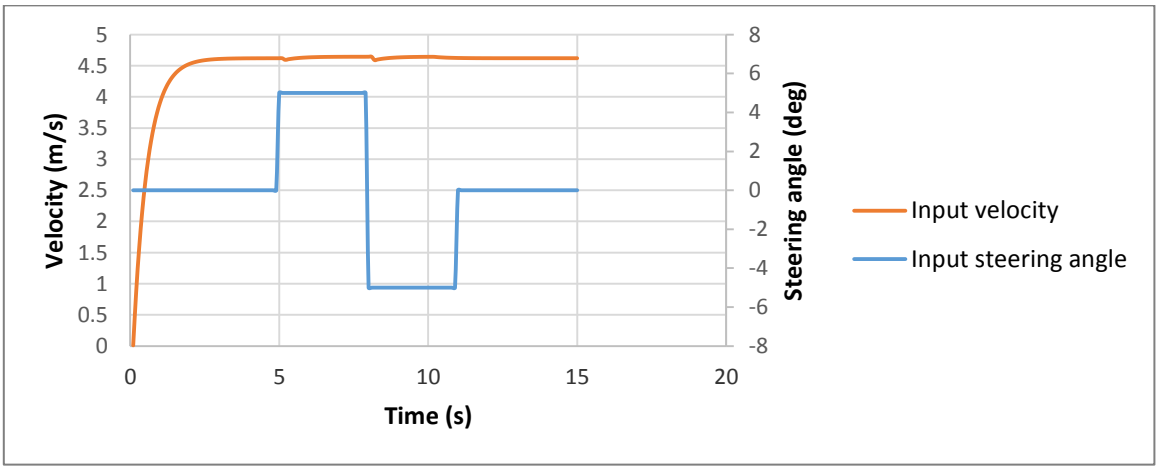

Figure 6.1: Showing velocity (maximum of 4.6 m/s) and steering angle (+5 and -5 degrees steps) inputs used during simulation 1 plotted against time on same graph. Longitudinal velocity is seen changing during step steering input.

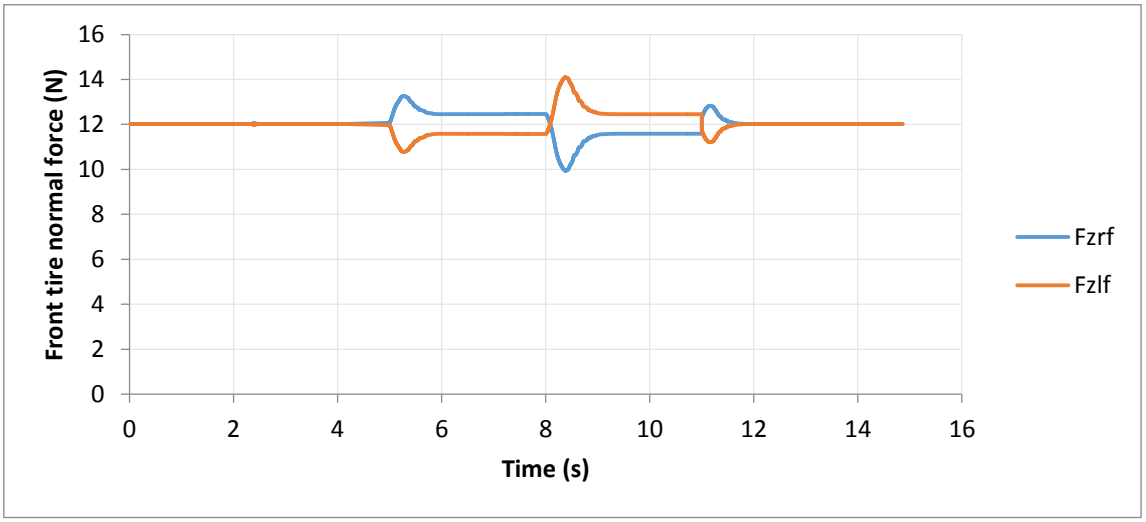

Figure 6.2: Showing right and left front tire normal forces (minimum values are Fzrf=9.9 N and Fzlf=10.8 N) from simulation 1 plotted against time on same graph.

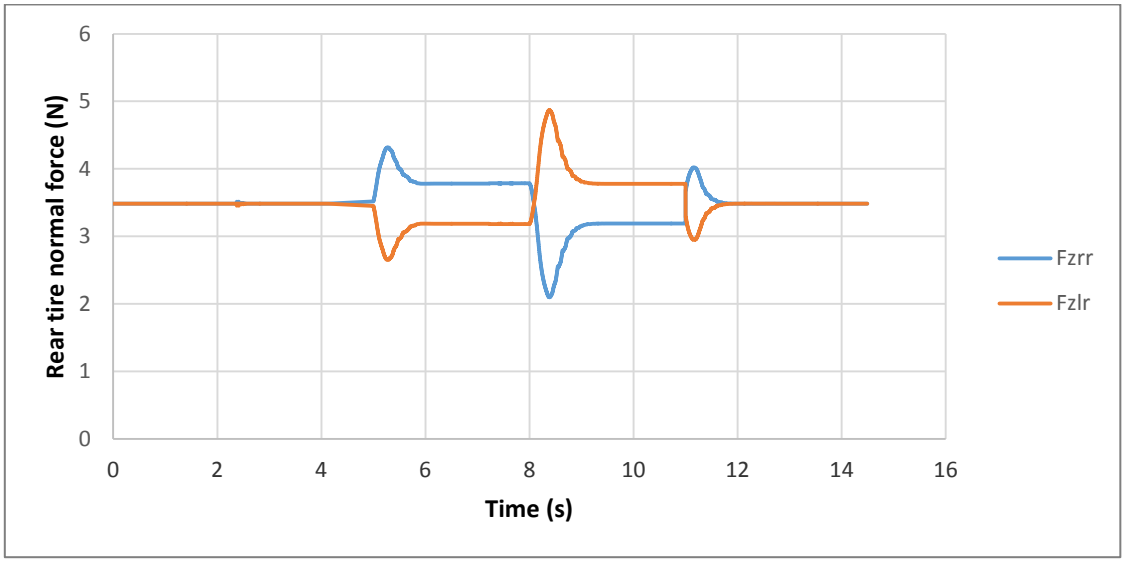

Figure 6.3: Showing right and left rear tire normal forces (minimum values are Fzrr=2.24 N and Fzlr=2.8 N) from simulation 1 plotted against time on same graph.

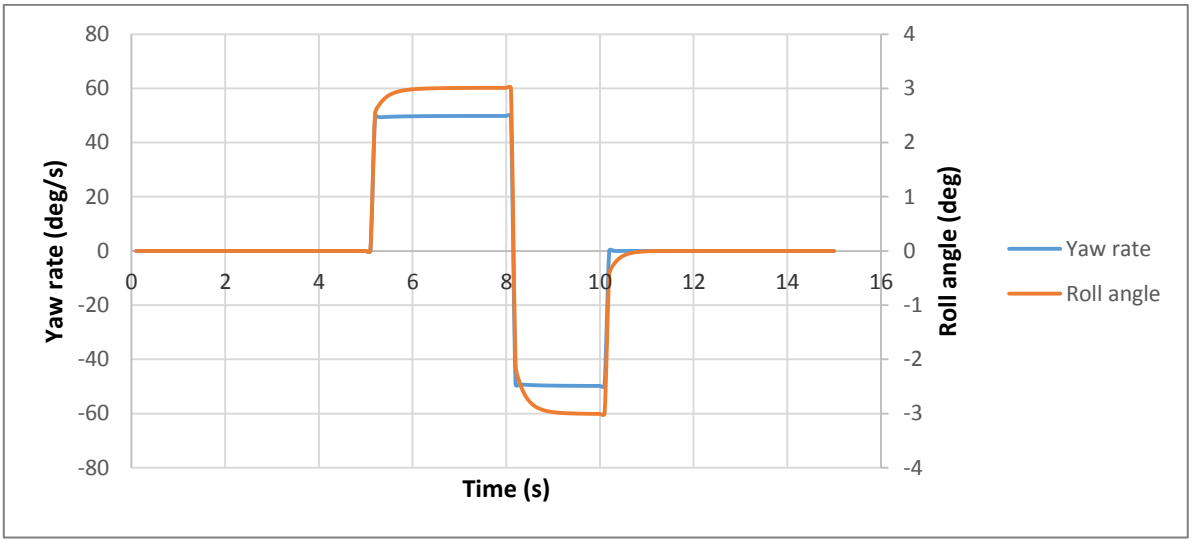

Figure 6.4: Showing roll angle (max. and min. values of +3 and -3 degrees respectively) and yaw rate (max. and min. values of +50 and -50 degrees/second respectively) for simulation 1 plotted against time on same graph.

It has been emphasised in literature that for most vehicles, it is the rear inside (in reference to turn) tire that will lift first in the event of rollover. The vehicle platform which has been used in this study has higher mass in front than the rear side (this can also be seen in the figures 6.2 and 6.3) hence the rear inside wheel is the first to lose contact in the event of rollover studied in this research. This being the case, and the fact that these simulations are done for sake of setting a limit on roll angle, the subsequent simulation results (graphs) are only showing normal forces for the rear tires. Normal forces for front tires are not plotted for brevity sake.

- **Simulation 2**

  (Velocity of 9.2 m/s and step steering angles of +5 and -5 degrees)

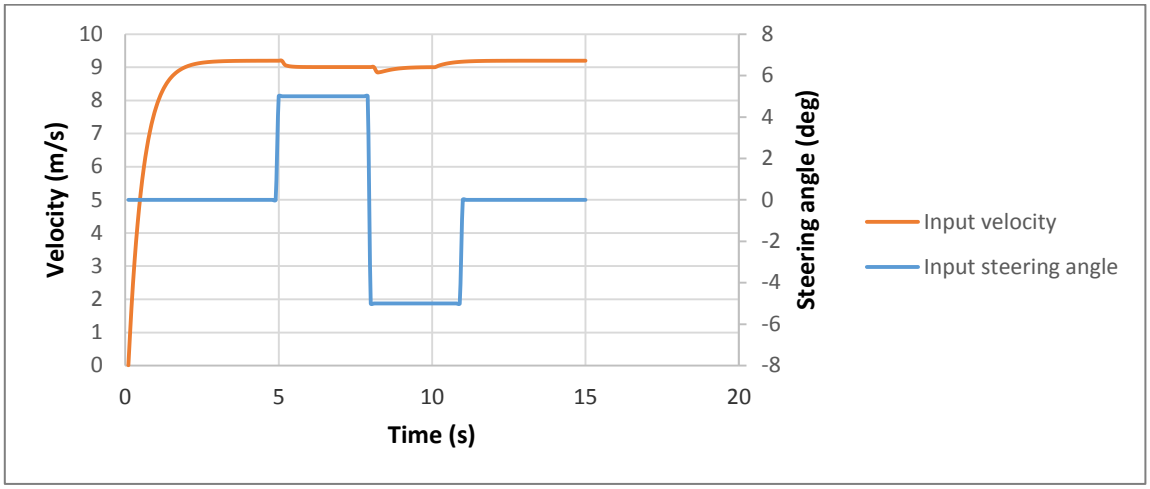

Figure 6.5: Showing velocity (maximum of 9.2 m/s) and steering angle (+5 and -5 degrees steps) inputs used during simulation 2 plotted against time on same graph. Longitudinal velocity is seen changing during step steering input.

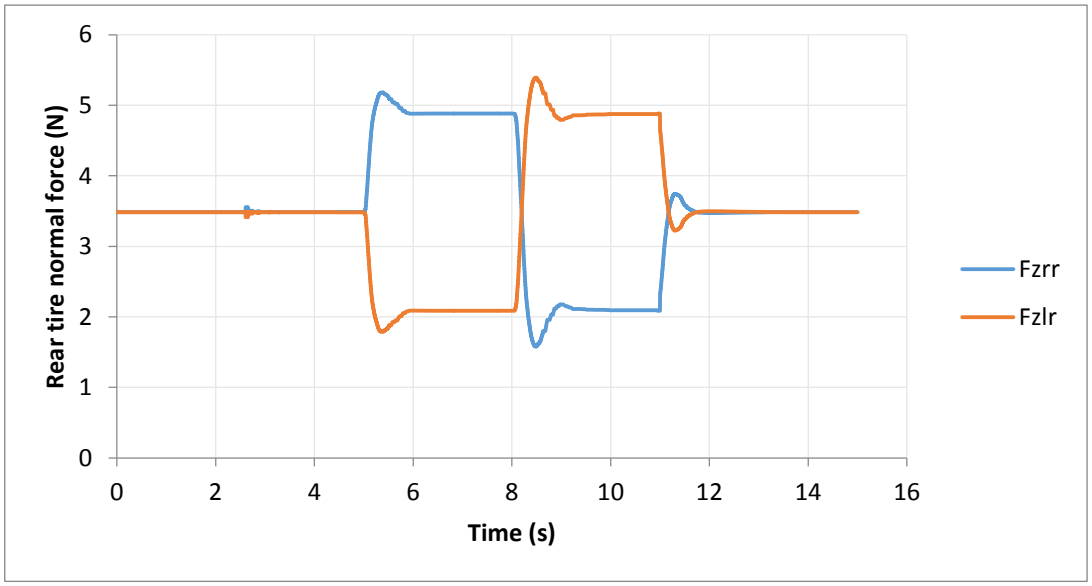

Figure 6.6: Showing right and left rear tire normal forces (minimum values are Fzrr=1.6 N and Fzlr=1.8 N) from simulation 2 plotted against time on same graph.

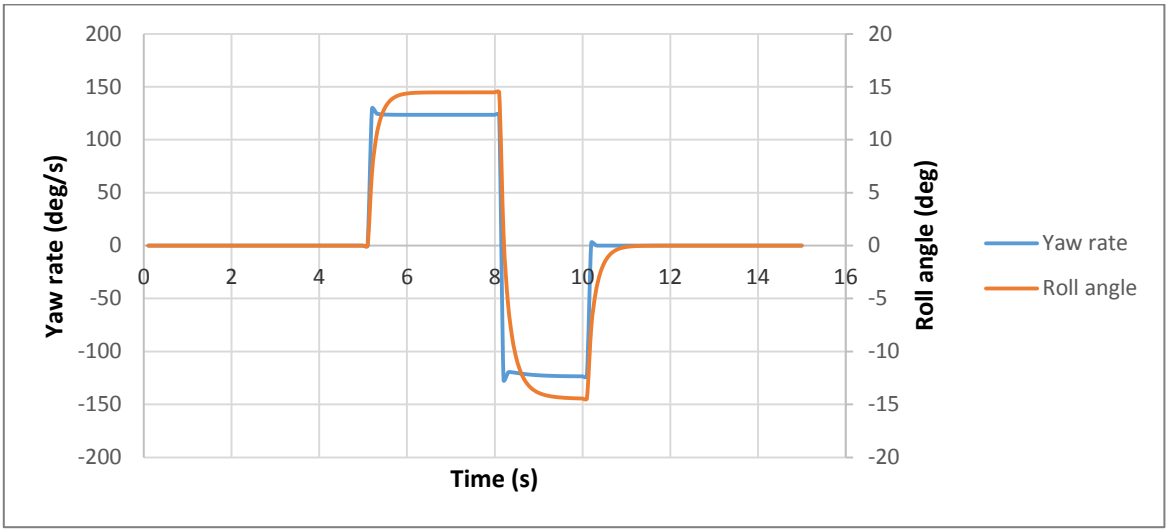

Figure 6.7: Showing roll angle (max. and min. values of +14.4 and -14.4 degrees respectively) and yaw rate (max. and min. values of +126 and -126 degrees/second respectively) for simulation 2 plotted against time on same graph.

- **Simulation 3**

  (Velocity of 9.2 m/s and step steering angles of +10 and -10 degrees)

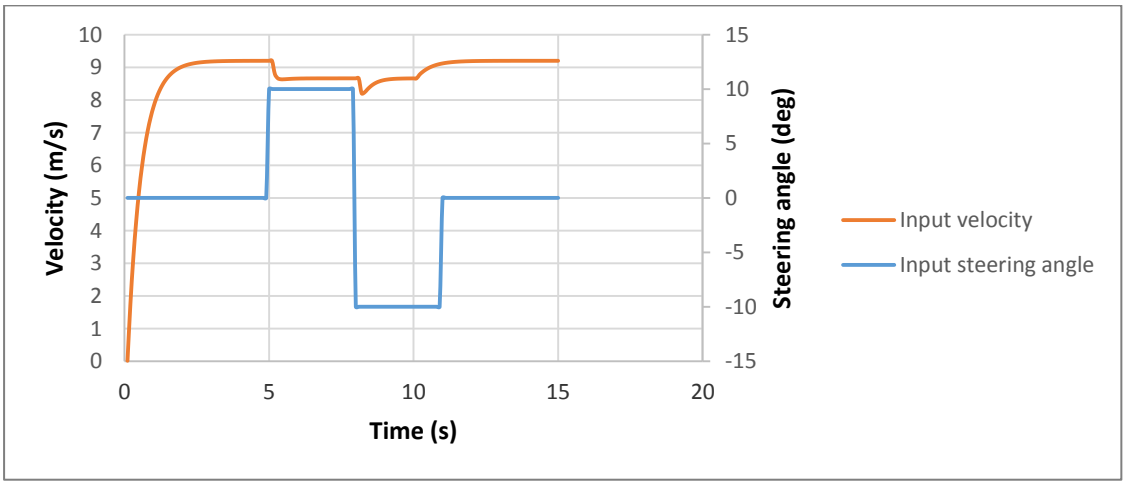

Figure 6.8: Showing velocity (maximum of 9.2 m/s) and steering angle (+10 and -10 degrees steps) inputs used during simulation 3 plotted against time on same graph. Longitudinal velocity is seen changing during step steering input.

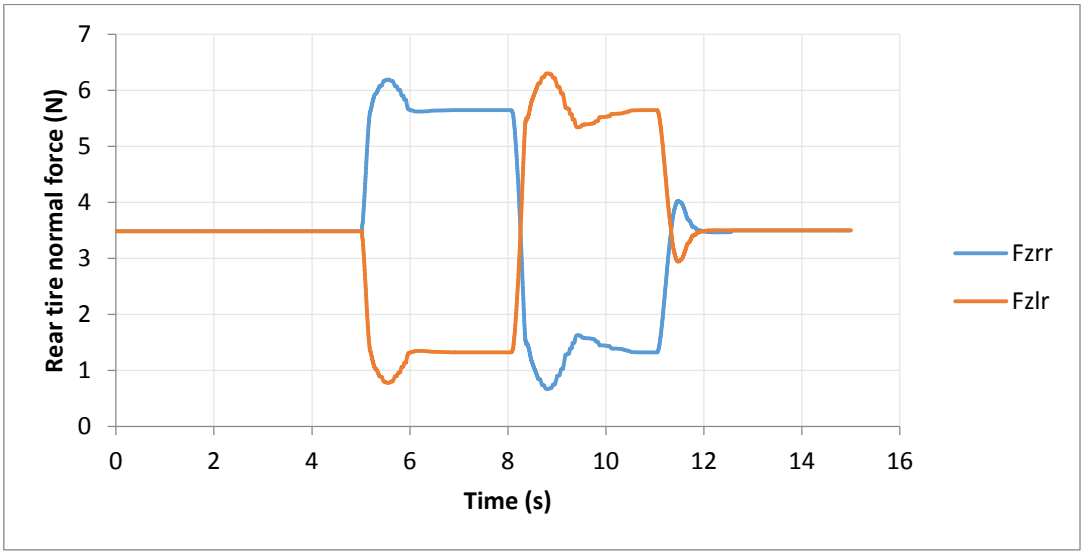

Figure 6.9: Showing right and left rear tire normal forces (minimum values are Fzrr=0.7 N and Fzlr=0.8 N) from simulation 3 plotted against time on same graph.

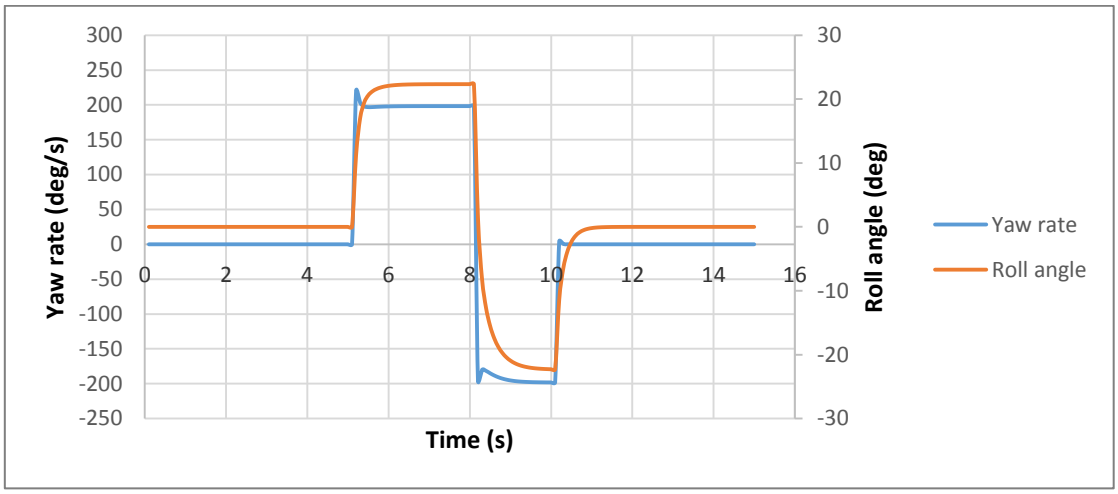

Figure 6.10: Showing roll angle (max. and min. values of +22 and -22 degrees respectively) and yaw rate (max. and min. values of +200 and -200 degrees/second respectively) for simulation 3 plotted against time on same graph.

Stability Limit

Looking at the three simulation scenarios (simulation 1, simulation 2, simulation 3), the load transfer ration (LTR) for each simulation can be calculated according to the formula given previously by equation 6.1. The formula is

$$LTR = \frac{F_{zR} - F_{zL}}{F_{zR} + F_{zL}}.$$

The formula was applied to the rear wheel normal forces to determine the LTRs, calculations and results have been tabulated as follows:

Table 6.2: Load Transfer Ratios (LTR) for the three simulation scenarios.

Simulation	Reference figure	Minimum LTR Value
Simulation 1	Figure 5.3	$\frac{2.24 - 4.728}{6.968} = -0.357$
Simulation 2	Figure 5.6	$\frac{1.6 - 5.368}{6.968} = -0.540$
Simulation 3	Figure 5.9	$\frac{0.7 - 5.998}{6.968} = -0.760$

Looking at the values in table 6.2, taking an LTR value of -0.760 (which is within the allowed values) would mean the vehicle will be able to move at very high velocity and perform steering manoeuvres (i.e. up to velocity of 9.2 m/s at a steering angle of ± 10 degrees) as can be seen from figure 6.8 under simulation 3. On the other side, taking the LTR of -0.357 would limit the vehicle to a maximum speed of 4.6 m/s at a steering angle of ± 5 degrees (figure 6.1). LTR of -0.540 is mid-way between these two extremes of the simulations performed. That is, it would limit the vehicle to a maximum speed of 9.2 m/s at a steering angle of ± 5 degrees (figure 6.5)

The vehicle platform that has been used in this study can only reach a maximum speed of around 6 m/s and a maximum steering angle of about ± 17 degrees as explained in system identification chapter. Taking this information into consideration and remembering that the main aim is to demonstrate the capability of the NMPC controller to track a given trajectory at the same time preventing the vehicle from rolling over by obeying constraints (roll angle mainly); it was decided to select an LTR that would likely activate constraints on the real system (physical platform).

Taking an LTR value of simulation 2 (-0.540) or simulation 3 (-0.760), both at velocity of 9.2 m/s, would reduce the chances of activating constraints (considering physical platform limitations). This is because the physical platform cannot reach that speed of 9.2 m/s. The LTR value of simulation 1 (-0.357) has higher chances of activating constraints hence can demonstrate the capability of the designed controller; therefore it is taken as a limit of stability in this study. In addition to that, steering angle is limited to ± 10 degrees for controller simulation. This has been done to keep the steering angle within its linear region of operation in reference to the physical platform used in this study. The relationship between steering input signal and resultant steering angle was observed to be nonlinear around extreme values.

From simulation 1, which has an LTR value of -0.357, it can easily be seen in figure 6.4 that this LTR translates into a maximum/minimum roll angle of ± 3 degrees and maximum/minimum yaw rate of ± 50 degrees per second. Therefore, these values of roll angle and yaw rate are taken as stability limit in this study. In addition to the yaw rate limit reached at here, reference yaw angle with respect to time was generated as part of

reference trajectory; this ensures that the vehicle does not spin out hence stabilizing the yaw rate naturally.

The defined roll angle limit results in putting more emphasis on roll angle (limit) than rollover in some parts of the chapters that follow. This is because the controller will be given roll angle constraint value which prevents the vehicle from rolling over. If the roll angle constraint is violated, it means rollover event has occurred.

Chapter 7

NMPC Controller Design

After determining stability limits of the scaled vehicle model used in this study, the next step is designing a controller for the vehicle. In this chapter therefore, the focus is on the design of a Nonlinear Model Predictive Control (NMPC) controller which can be implemented on a physical plant using the Automatic Control and Dynamic Optimisation (ACADO) Toolkit. The target implementation type is to implement within the ACADO environment; not the embedded type using code generation property of ACADO. The chapter starts by motivating the implementation within the ACADO environment approach and then finishes by designing the controller.

7.1 NMPC Formulation within ACADO

Model Predictive Control (MPC) is one of the problem classes that ACADO is designed for [54] as highlighted in literature review chapter of this study. Specifically, ACADO is designed targeting NMPC problems. When it comes to NMPC problems, ACADO has a *Process* class and a *Controller* class to simulate the problems. To achieve closed loop NMPC simulation, communication between *Process* and *Controller* is orchestrated by an instance of the *SimulationEnvironment* class which is also provided by ACADO. Figure 7.1 shows the *SimulationEnvironment* together with the *Process* and *Controller* in a closed loop scenario.

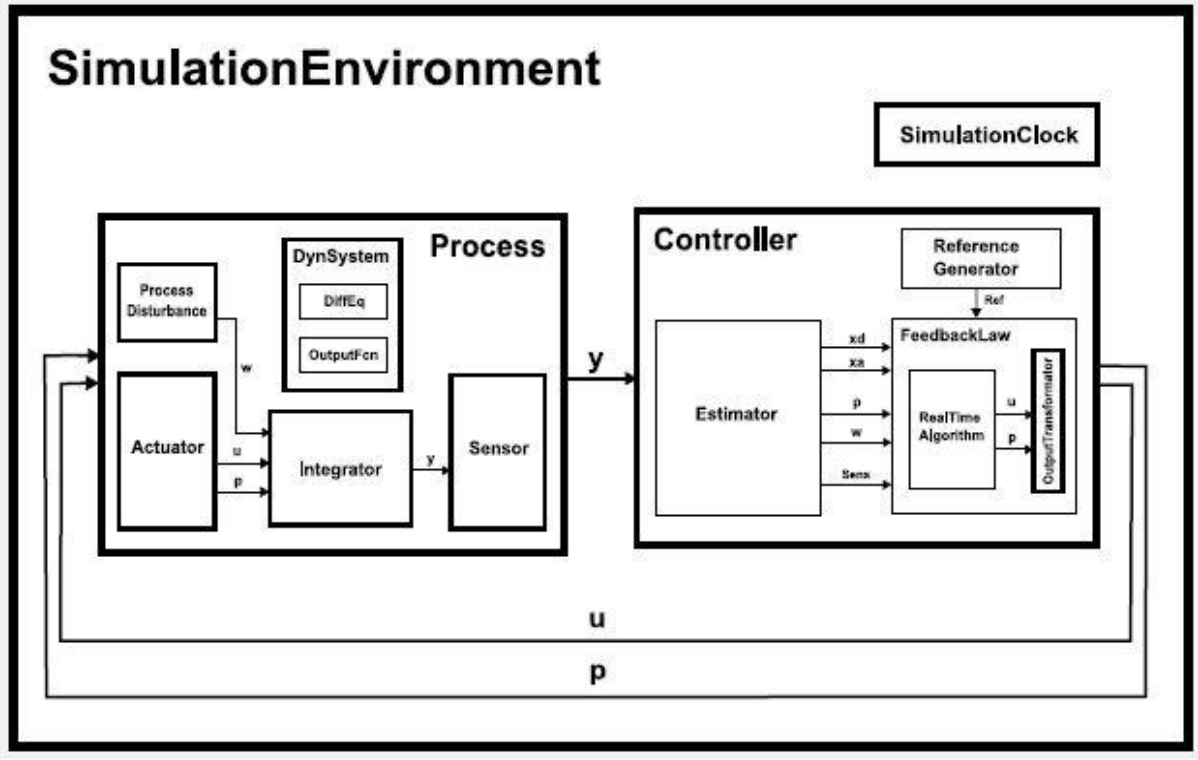

Figure 7.1: SimulationEnvironment, Process and Controller arrangement.

The Process class has as members a *dynamic system*, comprising a *differential equation* and an optional *output function*, modeling the process as well as an integrator capable of simulating these model equations. The simulation uses (optimized) control inputs from the controller.

The Controller class consists of three major blocks: first, an online *state/parameter estimator* which uses the outputs of the process to obtain estimates for the differential states or other parameters (implemented in code generation only at the writing of this report). Second, a *reference trajectory* can be provided to the control law. Finally, both the state/parameter estimates as well as the reference trajectory are used by the *ControlLaw* class to compute optimized control inputs. For NMPC purposes, the control law is a *RealTimeAlgorithm* based on a real-time iteration algorithm called Sequential Quadratic Programming (SQP). SQP details can be found in the literature review section of this report.

As seen in figure 7.1, as well as explained in the previous paragraphs, the structure shown in figure 7.1 allows only closed loop simulations of NMPC controllers. When the target is to implement the controller, ACADO provides Code generation tool which generates an optimised C-code of the controller which can be exported to an external embedded hardware. The generated code has shown to be efficient for millisecond nonlinear systems. The solution time is highly reduced. Examples of such implementation can be found in [69], [70].

In this study, an alternative implementation approach within the ACADO environment is investigated. This ACADO environment NMPC controller implementation form has some advantages over the code generation type of implementation. The approach has the following advantages:

- It makes it simple to debug the code; it eliminates the re-compiling and re-exporting the code to external hardware in case something does not work properly.
- It also makes it possible to link to external libraries in implementation, which is crucial in some applications.
- Operator can override the controller in case something goes wrong.
- It is a good approach for demonstration purposes to students to appreciate the working of ACADO in real time implementation, that is, it is possible to link to external libraries and have the real time data displayed in graphical interface.

The only big problem is the solution time, as it takes a bit longer to find optimal solution as compared to code generation approach. This can result in controller implementation on slower systems only. Taking into consideration the fact that microprocessor technology is advancing in terms of increasing microprocessor speed and other features, the approach is promising for even faster systems.

7.2 NMPC Approach

To use the proposed approach within ACADO environment, simulation environment class shown in figure 7.1 has been used in conjunction with the actual plant. The new representation of the whole arrangement can be viewed as in figure 7.2.

In figure 7.2, y1 and u1 are representing process1 (it was named 'process' in figure 7.1) output and control action respectively within the simulation environment. In addition to the process1 class inside the simulation environment, another process2 class has been defined outside the simulation environment, with output y2, as can be seen in the figure 7.2. The procedure of operation can be explained as follows: Normally (without considering process2) when the simulation environment is performing closed loop simulation, the environment is initialised with initial process1 states at the beginning of the simulation. After that it keeps running the simulation in closed form up to the end of the defined simulation time. During this running time, the output of process1 (y1) are used to initialise the controller which then solves to find optimal control actions. The first of these optimised control actions is written to the process1 which integrates to produce new y1, and the cycle repeats until end of defined time.

Now, with this new arrangement (considering process2), the *SimulationEnvironment* class is not allowed to simulate process1 and controller in closed loop. Instead of writing control actions to process1 as described in the previous paragraph, the control actions are written to external process denoted process2 in the diagram at each sampling time. The external process2 integrates using the optimised control action (u2) from the controller and its own previous output to produce new output y2. This new output is then used to initialise the simulation environment, hence the controller. This way, it can be seen that the loop is closed using the external process2 (solid lines on the figure 7.2). This cycle repeats up until end of defined total running time.

It can be seen from explanation of operation that process1 does not play a crucial role in this new approach. Actually the u1 (control action) going to process1 and y1 (process1 output) output as shown in figure 7.2 (dotted lines) are not used in this approach. Both of these (u1 and y1) and process1 still remain in the diagram (as well as in actual implementation) just to keep the *SimulationEnvironment* class operational; this is because process1 is one of the input arguments of the class.

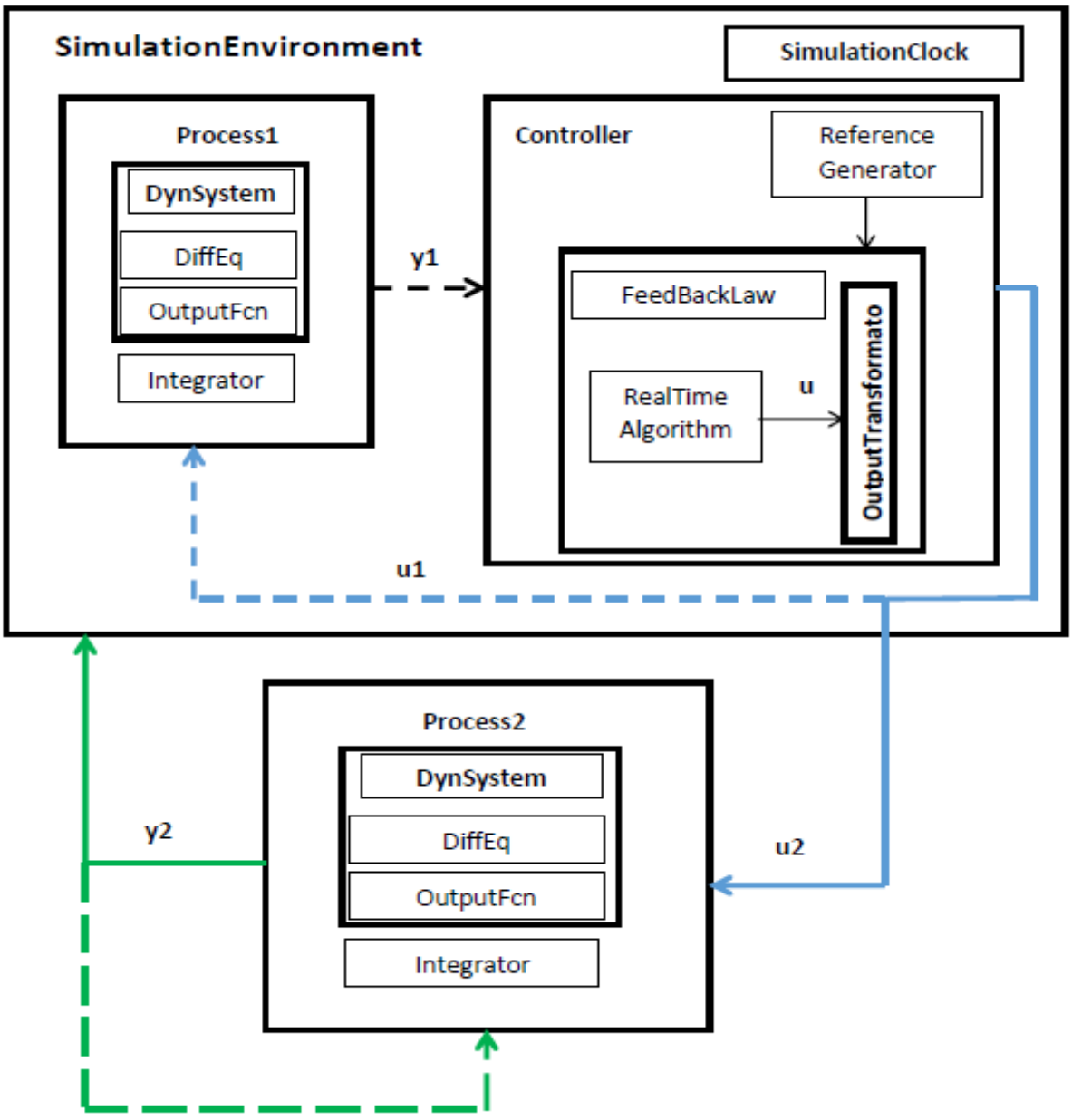

Figure 7.2 Implementation of the NMPC controller using SimulationEnvironment class of ACADO using the proposed approach.

Configuring the simulation environment in this way, allows implementation of the controller by just removing the external process (denoted process2) in simulation, and replacing with a real plant. It is also possible to estimate states of the plant and even augment if there is need, by just putting an estimator between the external process and the simulation environment. This augmentation has been used in implementation of the proposed NMPC controller form on a DC motor in this study. More details of DC motor implementation are in appendix C.

7.3 NMPC Controller

In this study, the problem has been cast as tracking MPC. The problem has been formulated in least squares sense, where the square of the error between the reference trajectory of output and the actual output is penalised. In addition to that, the square of the difference between the reference control action and actual control action is penalised as well. The weighting parameters play an important role in determining where to put more emphasis between the two. Mathematically, the NMPC controller in this study solves the following problem:

$$\underset{\xi(.),\ u(.)}{\text{minimise}} \quad \int_0^T \left(\left\| y(t) - y_{ref}(t) \right\|_Q^2 + \left\| u(t) - u_{ref}(t) \right\|_R^2 \right) dt \,, \tag{7.1a}$$

$$\text{subject to} \quad \xi(0) = \xi_0 \,, \tag{7.1b}$$

$$\forall t \in [0, T]: \quad \dot{\xi}(t) = f\left(t, \xi(t), u(t)\right), \tag{7.1c}$$

$$\forall t \in [0, T]: \quad y(t) = h\left(t, \xi(t), u(t)\right), \tag{7.1d}$$

$$\forall t \in [0, T]: \quad \underline{u} \leq u(t) \leq \bar{u}, \tag{7.1e}$$

$$\forall t \in [0, T]: \quad \underline{\xi} \leq \xi(t) \leq \bar{\xi}. \tag{7.1f}$$

Equation 7.1a represents the objective function of the controller. In this equation, T is the fixed length of the prediction horizon of the NMPC controller and $t \in \mathbb{R}$ is time. $\mathbb{R}^n$ is used to denote the set of real-valued n-vectors. $y \in \mathbb{R}^p$ is the output of process2 as depicted in figure 7.2. y_{ref} is the process2 output reference trajectory to be given to the controller. $u \in \mathbb{R}^m$ is the input and u_{ref} is the input reference. The norms in the objective function have been weighted with positive-definite matrices Q and R for output and input respectively.

Equations 7.1c and 7.1d represent process2 state equations and output equations respectively. They have been denoted by f and h for state and output equations respectively. In these equations, $\xi \in \mathbb{R}^n$ is representing the state of process2, u is the input and t is time.

Equation 7.1b represents the initial condition of process2. The initial condition specifies the value of the state ξ at time $t = t_0$, and a solution is searched to the controller equation for time greater than t_0, $t \in \mathbb{R}_{\gg 0}$. Often the initial time is defined to be zero, with a corresponding initial condition, in which case $t \in \mathbb{R}_{\gg 0}$.

Equations 7.1e and 7.1f represent constraints on process2 input and states respectively. Upper and lower bounds have been denoted by $\underline{u}, \bar{u}$ and $\underline{\xi}, \bar{\xi}$ for the inputs and states, respectively. These constraints are to be respected by the NMPC controller.

ACADO software implements real-time algorithm to solve the NMPC controller defined in equation 7.1. For discretization of the equation, both single shooting and multiple shooting are implemented [54]; multiple shooting is used in this study. At each time step t, the objective function is optimized under the constraints starting from the initial state to obtain an optimal control sequence. The first of such optimal moves is the control action applied to the process2 at time t. At time $t + 1$, a new optimization is solved over a shifted prediction horizon starting from the new measured state. The time interval between time step t + 1 and time step t is the called the sampling time.

Chapter 8

NMPC Controller for the Vehicle Model

This chapter focuses on closed loop simulation of the NMPC controller form designed in chapter 7 with the full nonlinear vehicle model that has been derived in chapter 4 of this report. These simulations come after the proposed and designed NMPC controller showed good performance when it was successfully simulated and implemented on a DC motor system for velocity and position control. The results indicated that the controller is capable of achieving set point tracking as well as respecting constraints. More details on the DC motor control are in the appendix C of this report.

The aims of the simulations in this chapter are to check whether the controller is able to track a given trajectory as well as respecting the constraints when applied to the vehicle model developed. In this study, the constraints are mainly roll angle and yaw angle (thereby yaw-rate). As previously stated, violation of roll angle constraint means rollover event has occurred. In addition to tracking and respecting constraints; robustness and disturbance rejection of the designed controller are also investigated. The chapter starts by doing closed loop simulation with the vehicle model, then robustness and disturbance rejection analysis follow, and finally looks at tuning parameters of NMPC in conjunction with feasibility of the controller.

8.1 NMPC Controller Simulation

This first section presents simulation results of the NMPC controller designed in chapter 6 of this report. The controller was simulated in closed loop form with the scaled vehicle dynamic model which was derived in chapter 4 and modified during system identification in chapter 5. For continuity purposes, the NMPC controller equation is presented here again as follows:

$$\underset{\xi(.),\ u(.)}{\text{minimise}} \quad \int_0^T \left(\left\| y(t) - y_{ref}(t) \right\|_Q^2 + \left\| u(t) - u_{ref}(t) \right\|_R^2 \right) dt \tag{8.1a}$$

$$\text{subject to} \quad \xi(0) = \xi_0 \,, \tag{8.1b}$$

$$\forall t \in [0, T]: \quad \dot{\xi}(t) = f\left(t, \xi(t), u(t)\right), \tag{8.1c}$$

$$\forall t \in [0, T]: \quad y(t) = h\left(t, \xi(t), u(t)\right), \tag{8.1d}$$

$$\forall t \in [0, T]: \quad \underline{u} \leq u(t) \leq \bar{u}, \tag{8.1e}$$

$$\forall t \in [0,T]: \qquad \underline{\xi} \leq \xi(t) \leq \overline{\xi}. \tag{8.1f}$$

The variables in this equation 8.1 have already been explained in chapter 6 of this report.

The nonlinear state space vehicle model which was used in system identification chapter was also used in this chapter to simulate the controller. The vehicle model has already been presented in previous chapters. In this chapter, only the state equations have been rewritten to ensure continuity and brevity. The state equations are written as follows:

$$M\dot{U}_{ox} = \Sigma F_x + \left(M_{uf}l_f - M_{ur}l_r\right)\dot{\psi}^2 - 2h_{rc}M\,\dot{\psi}\dot{\phi} + M\,\dot{\psi}U_{oy}, \tag{8.2a}$$

$$M\dot{U}_{oy} = \Sigma F_y + \left(M_{ur}l_r - M_{uf}l_f\right)\ddot{\psi} + h_{rc}M\ddot{\phi} - M\,\dot{\psi}U_{oy} - M\,\dot{\psi}U_{ox}, \tag{8.2b}$$

$$\begin{aligned} I_{zz}\ddot{\psi} = {} & \left(F_{ylf} + F_{yrf}\right)l_f - \left(F_{ylr} + F_{yrr}\right)l_r + \left(F_{xlf} + F_{xrf}\right)\frac{t_f}{2} + \left(F_{xlr} - F_{xrr}\right)\frac{t_r}{2} \\ & + \left(M_{ur}l_r - M_{uf}l_f\right)\left(\dot{U}_{oy} + \dot{\psi}U_{ox}\right) - I_{xz}\ddot{\phi}, \end{aligned} \tag{8.2c}$$

$$\begin{aligned} & \left(I_{xx} + Mh_{rc}^2\right)\ddot{\phi} + I_{xz}\ddot{\psi} \\ & \quad = Mgh_{rc}\phi - \left(k_{\phi f} + k_{\phi r}\right)\phi - \left(b_{\phi f} + b_{\phi r}\right)\dot{\phi} + h_{rc}M\left(\dot{U}_{oy} + \dot{\psi}U_{ox}\right), \end{aligned} \tag{8.2d}$$

$$I_w\dot{\omega}_{f,l} = -F_{lf,l}r_w + T_{f,l} - b_w\dot{\omega}_{f,l}, \tag{8.2g}$$

$$I_w\dot{\omega}_{f,r} = -F_{lf,r}r_w + T_{f,r} - b_w\dot{\omega}_{f,r}, \tag{8.2h}$$

$$I_w\dot{\omega}_{r,l} = -F_{lr,l}r_w + T_{r,l} - b_w\dot{\omega}_{r,l}, \tag{8.2i}$$

$$I_w\dot{\omega}_{r,r} = -F_{lr,r}r_w + T_{r,r} - b_w\dot{\omega}_{r,r}, \tag{8.2j}$$

$$\dot{Y} = \dot{x}\,sin\,\Psi + \dot{y}\,cos\,\Psi, \tag{8.2e}$$

$$\dot{X} = \dot{x}\,cos\,\Psi - \dot{y}\,sin\,\Psi. \tag{8.2f}$$

As explained in previous chapters of this report, the inputs to the vehicle platform used in this study were torque (T) produced by the driving motor and front wheel steering angle (δ_f) . To have smoother velocity profiles in both x and y directions (ride comfort), the rates of change of inputs were taken as inputs to the model rather than the actual inputs. This is usually referred to as velocity form in literature. Adopting this velocity form for the two controls, results in addition of two states to the state equations presented in equation 8.2. The two new states are the torque and steering angle which are written as follows:

$$\dot{T} = \Delta T, \tag{8.2g}$$

$$\dot{\delta_f} = \Delta\delta_f. \tag{8.2h}$$

In relation to the controller equation (equation 8.1), these state equations (equation 8.2) were compactly written as

$$\dot{\xi}(t) = f\big(t, \xi(t), \Delta u(t)\big). \tag{8.3}$$

Where $\xi(t) \in R^n$ represented the states of the system and $\Delta u(t) \in R^m$ represented the inputs, $n = 14$ is the number of states and $m = 2$ is the number of inputs. The fourteen states are longitudinal and lateral velocities in the body frame, the yaw angle, yaw rate, the roll angle, roll rate, lateral and longitudinal vehicle coordinates in the inertial frame, the angular velocity on the four wheels and the two states of inputs. These are denoted respectively as $\xi = [\,\dot{y},\ \dot{x}, \Psi, \dot{\Psi}, \Phi, \dot{\Phi}, Y, X, \omega_{f,l}, \omega_{f,r}, \omega_{r,l}, \omega_{r,r}, T, \delta_f\,]'$. Where $\omega_{f,l}, \omega_{f,r}, \omega_{r,l}$ and $\omega_{r,r}$ denotes the angular velocity on the front left, front right, rear left and rear right wheels, respectively. The two inputs are $\Delta u = [\Delta T, \Delta\delta_f]'$ where $\Delta\delta_f$ denotes the rate of change of front steering angle, and ΔT is the rate of change of braking or tractive torque on the four wheels. Positive ΔT denotes driving torque rate while negative ΔT denotes braking torque rate.

Recalling from chapter 6 of this report where stability limits were defined for the vehicle platform, the limit for the values are presented again here for use by the controller as follows:

- Maximum/minimum roll angle values: ± 3 degrees (± 0.05 radians).
- Maximum/minimum yaw rate values: ± 50 degrees per second (± 0.87 radians per second.

Similarly, it was mentioned during system identification that the physical platform had limits in terms of steering angle and input. These limits are captured here again for completeness as follows:

- Maximum/minimum driving torque: ± 1 Newton meter (Nm).
- Maximum/minimum steering angle: ± 10 Degrees (± 0.175 Radians).

The limits on driving torque rate ΔT and steering angle rate $\Delta\delta$ were reached at upon doing a number of simulations and selecting those which produced better results. In implementations they are usually limited by considering actuator limit as well. They were limited as follows:

- Maximum/minimum driving torque rate: ± 8 Newton meter per second (Nm/s).
- Maximum/minimum steering angle rate: ± 1 radians per second (rads/s).

To test the capability of the designed controller in both tracking and rollover prevention by respecting constraints, it was necessary to design a trajectory that is complex enough so that lateral vehicle dynamics are exited. The trajectory designed defines lateral vehicle position in inertia (global) frame (Y) as a function of longitudinal vehicle position in inertia frame (X). As highlighted in the introduction chapter, in addition to path following and rollover prevention, spinning out avoidance was also of concern in this study. Spinning out is a function of yaw rate, which has partly already been taken care by constraining the yaw rate.

Furthermore, the yaw angle reference trajectory was defined which further constrained the yaw rate as well. The yaw angle reference trajectory was defined as the derivative of Y position with respect to X position. The Y as function of X trajectory was defined as follows:

$$\begin{aligned} Y_{ref} = \left(\frac{6.05}{2}\right) * \left(1 + \tanh\left(\left(\frac{5}{25}\right) * \left(X_{ref} - 18.19\right) - 1.2\right)\right) - \left(\frac{8.7}{2}\right) \\ * \left(1 + \tanh\left(\left(\frac{5}{21.95}\right) * \left(X_{ref} - 26.46\right) - 1.2\right)\right) + \left(\frac{2.7}{2}\right) \\ * \left(1 + \tanh\left(\left(\frac{8}{25}\right) * \left(X_{ref} - 40.19\right) - 1.2\right)\right). \end{aligned} \tag{8.4}$$

The yaw angle reference trajectory was then defined as

$$\Psi_{ref} = \frac{dY_{ref}}{dX_{ref}}. \tag{8.5}$$

Finally, after defining all the necessary variables, the controller equation presented by equation 8.1 was rewritten mathematically as follows:

$$\underset{\xi(.),\ u(.)}{\text{minimise}} \quad \int_0^T \left(\left\| y(t) - y_{ref}(t) \right\|_Q^2 + \left\| \Delta u(t) - \Delta u_{ref}(t) \right\|_R^2 \right) dt \tag{8.6a}$$

$$\text{subject to} \quad \xi(0) = 0, \tag{8.6b}$$

$$\forall t \in [0,T]: \quad \dot{\xi}(t) = f\left(t, \xi(t), u(t)\right), \tag{8.6c}$$

$$\forall t \in [0,T]: \quad y(t) = \xi(t), \tag{8.6d}$$

$$\forall t \in [0,T]: \quad -8.0 \leq \Delta T(t) \leq 8.0, \tag{8.6e}$$

$$\forall t \in [0,T]: \quad -1.0 \leq \Delta\delta_f(t) \leq 1.0, \tag{8.6f}$$

$$\forall t \in [0,T]: \quad -1.0 \leq T(t) \leq 1.0, \tag{8.6g}$$

$$\forall t \in [0,T]: \quad -0.175 \leq \delta_f(t) \leq 0.175, \tag{8.6h}$$

$$\forall t \in [0,T]: \quad -0.05 \leq \Phi(t) \leq 0.05, \tag{8.6i}$$

$$\forall t \in [0, T]: \qquad -0.87 \leq \dot{\Psi}(t) \leq 0.87. \tag{8.6j}$$

The reference trajectories were presented as

$$y_{ref} = \left[0, 0, \Psi_{ref}, 0,0,0, Y_{ref}, X_{ref}, 0, 0, 0, 0, 0,0 \right],$$

$$\Delta u_{ref} = [0, 0].$$

The remaining variables were defined as follows: t was in steps of $\frac{0.5}{5}$ seconds, $T = 0.5$ seconds, $Q = \text{diagonal}[\ 1.0\text{e}-4, 1.0\text{e}-8,\ 1.0\text{e}-1,\ 1.0\text{e}-8, 1.0\text{e}-8,\ 1.0\text{e}-8,\ 1.0\text{e}-1,\ 1.0\text{e}-1,\ 1.0\text{e}-8, 1.0\text{e}-8,\ 1.0\text{e}-8, 1.0\text{e}-8, 1.0\text{e}-8]$ and $R = \text{diagonal}[1.0\text{e}-3,\ 1.0\text{e}-3]$. The sampling time used was 0.025 seconds.

In this study it is assumed that there is 1 meter trajectory width clearance (both sides) for full scale vehicle (typical value for most roads). This means that if a vehicle deviates by more than 1 meter from the reference trajectory, then it is outside the defined path. Recalling that a 1/10th scaled vehicle is used, 1 meter clearance is scaled to 0.1 meters clearance as well. Therefore, more than 0.1 meters deviation from the reference trajectory is interpreted as outside defined trajectory.

For clarity and easy interpretation of the results, all the simulation results have been plotted against X position coordinate of inertia frame. This idea brings uniformity in results presentation and makes it easy to relate points of interest as seen in the results. The reference Y_{ref} trajectory and actual Y trajectory followed by the controller have been plotted on the same graph, similarly the reference Ψ_{ref} and actual Ψ have also been plotted on the same graph. After these graphs, then steering angle, longitudinal velocity, roll angle, driving torque graphs are plotted as well. Finally the yaw rate, steering rate and driving torque rate graphs are plotted.

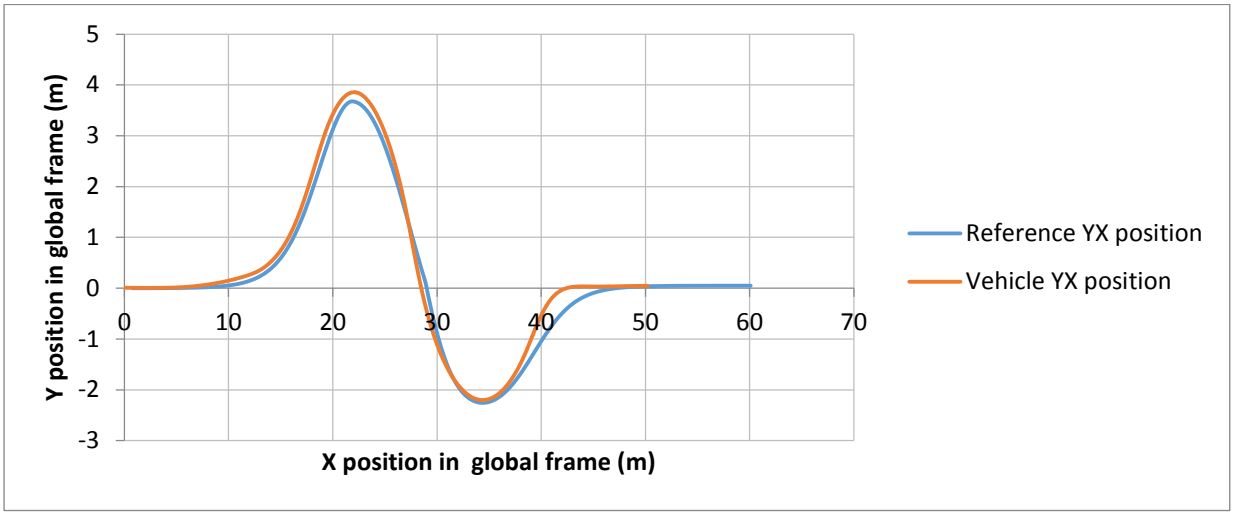

Figure 8.1: Showing reference YX position and actual YX position trajectories plotted against X position in inertia (global) coordinate frame. The vehicle is seen tracking the reference trajectory, the vehicle went outside the trajectory by 0.5 m at 41 m of X position.

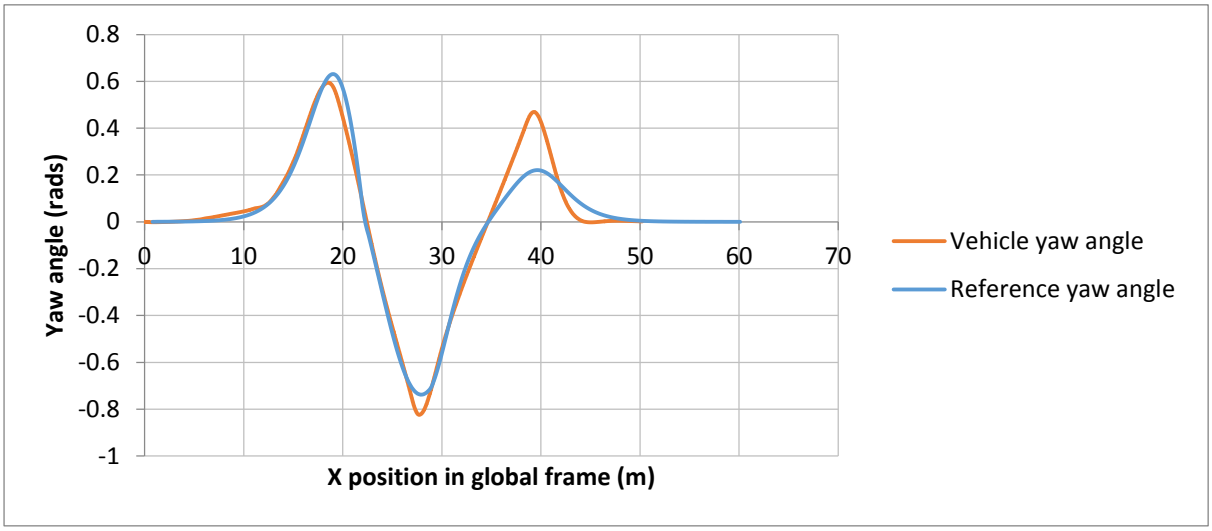

Figure 8.2: Showing reference yaw angle trajectory and actual yaw angle trajectories plotted against X position in inertia coordinate frame.

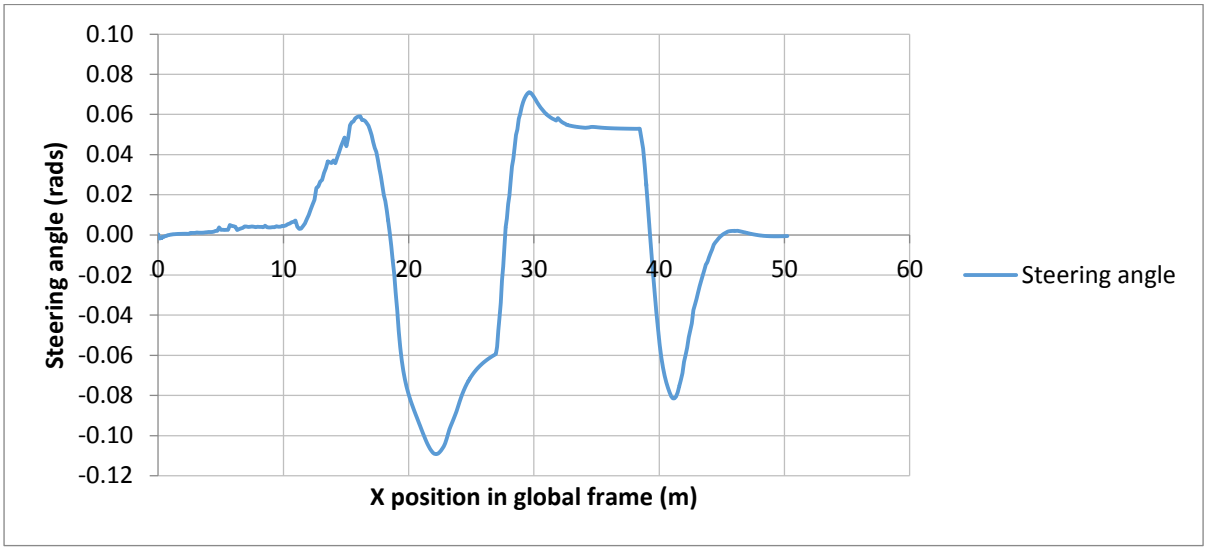

Figure 8.3: Showing vehicle steering angle plotted against X position, the steering angle changes to track reference YX position accordingly.

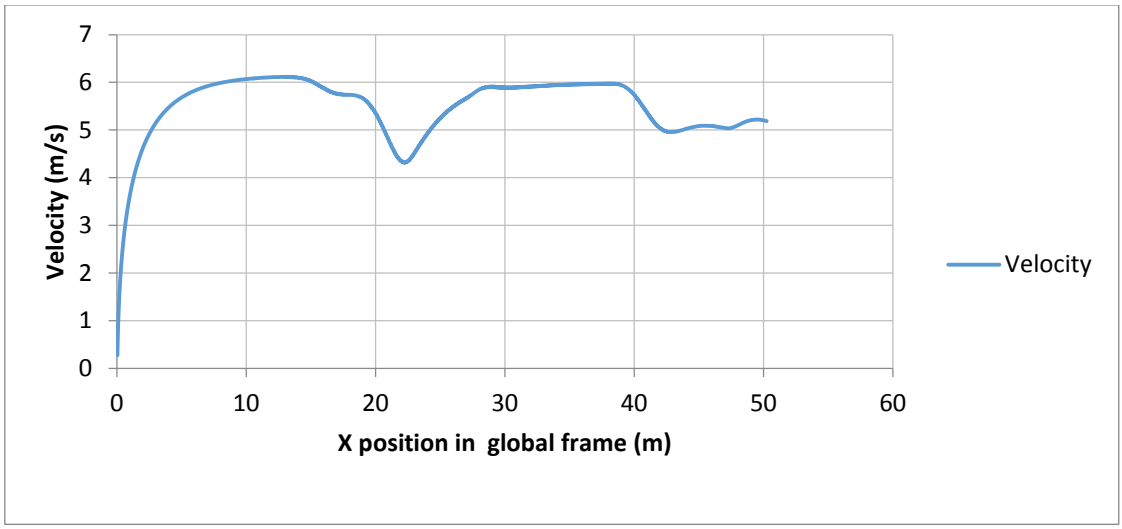

Figure 8.4: Showing vehicle longitudinal velocity plotted against X position, velocity increases to maximum value then decreases due to active constraint at two points.

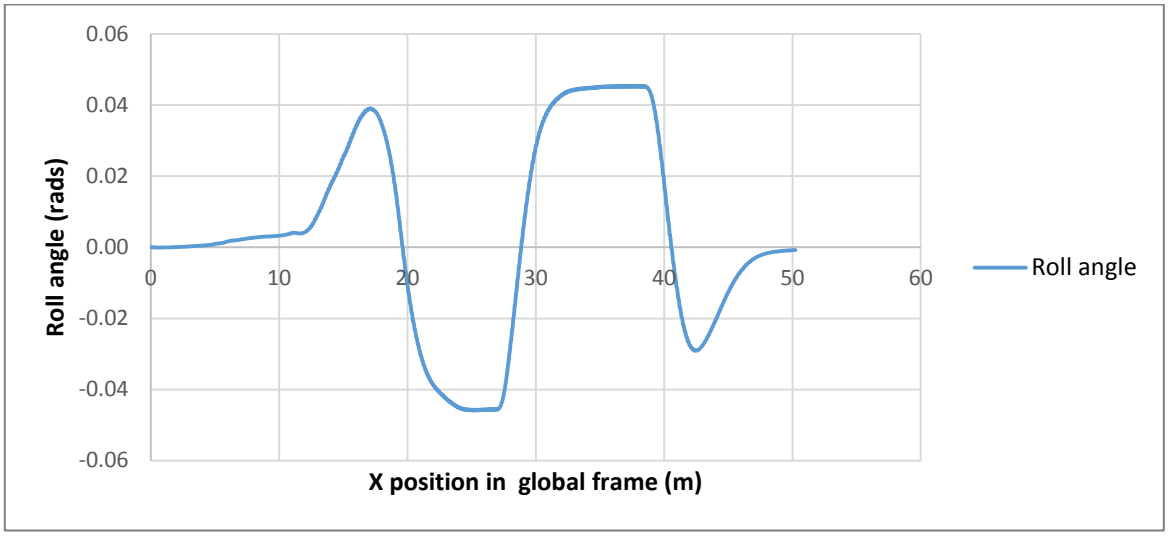

Figure 8.5: Showing vehicle roll angle plotted against X position, roll angle is seen changing with changing reference trajectory.

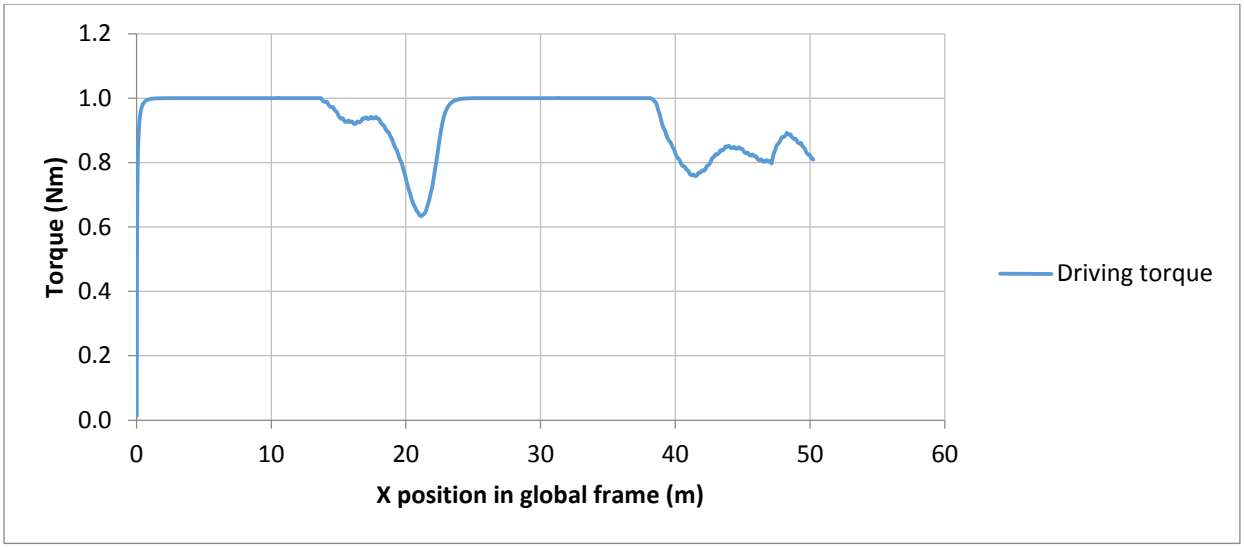

Figure 8.6: Showing vehicle driving torque plotted against X position, torque reached maximum value (saturation) and decreased at two points due to roll angle constraint being active.

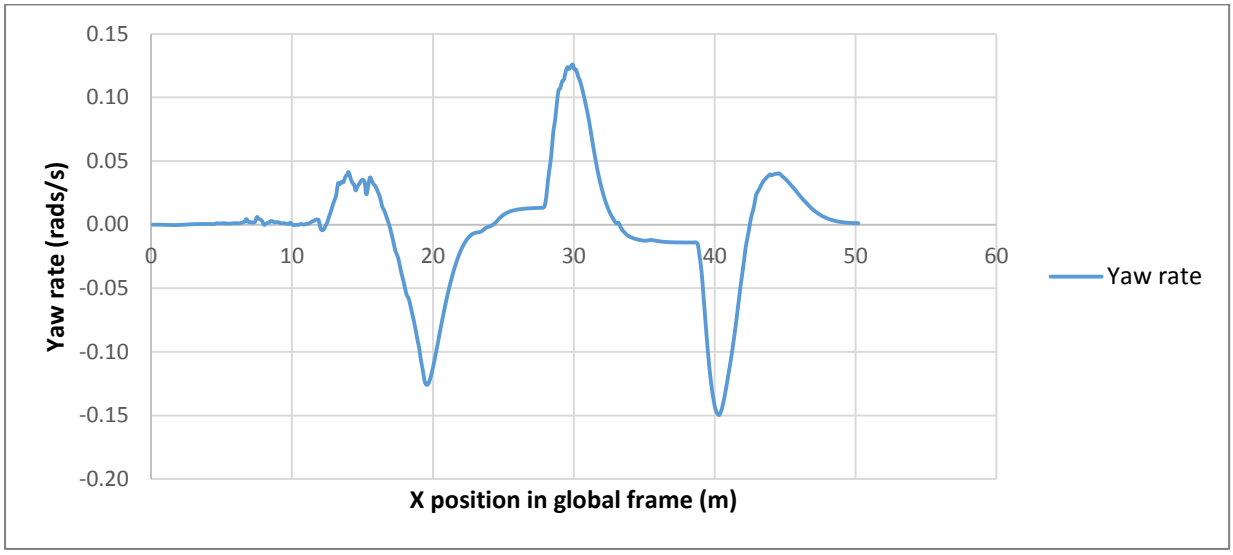

Figure 8.7: Showing vehicle yaw rate plotted against X position, yaw rate is seen changing with changing reference YX trajectory and did not reach its maximum value allowed.

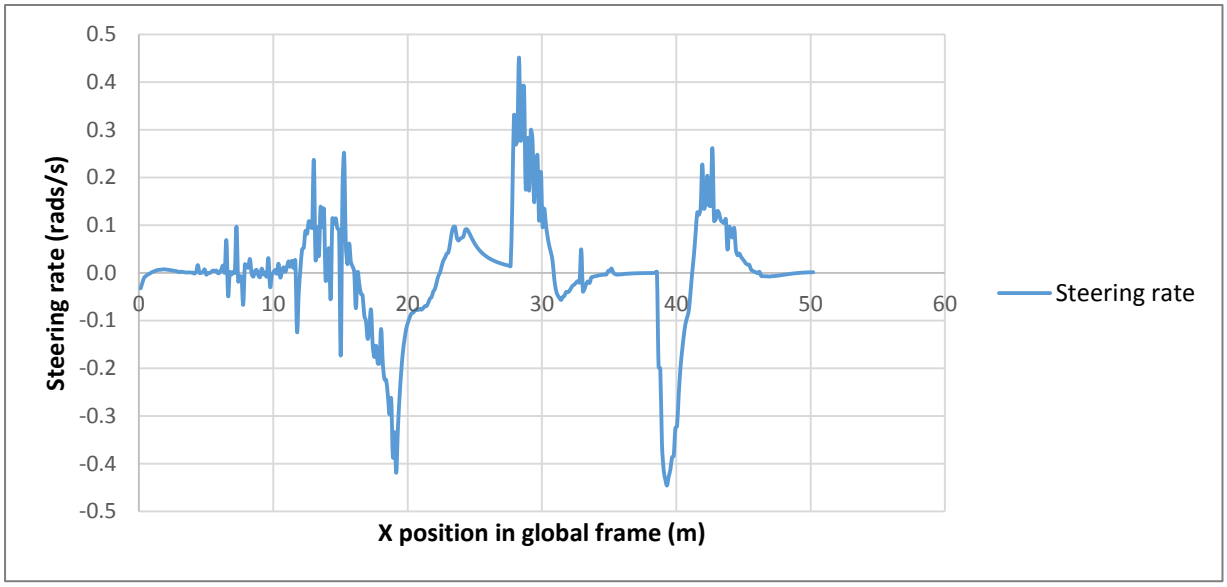

Figure 8.8: Showing vehicle steering rate plotted against X position, steering angle rate did not reach its maximum allowed value.

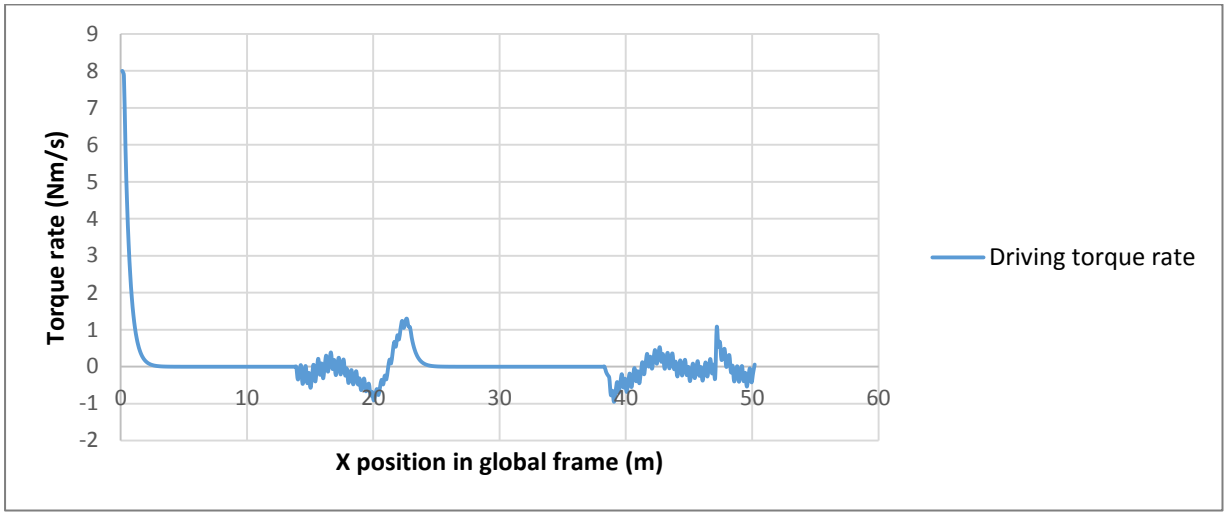

Figure 8.9: Showing vehicle driving torque rate plotted against X position, torque rate is seen reaching its maximum allowed value at beginning of simulation. Driving torque rate constraint was only active during simulation starting time.

From figure 8.1, it is seen that the controller is able to track the given Y position reference trajectory with errors less than 0.1 meters. Maximum error is seen to be 0.5 meters at 41 meters of X position. Similar situation is seen in figure 8.2 where reference yaw angle has been plotted against actual vehicle yaw angle, the controller shows a fairly good tracking behaviour of the reference. Figures 8.3 and 8.4 show vehicle steering angle and longitudinal velocity respectively; the steering angle is seen changing in synchronous to the changing Y position trajectory in figure 8.1. Longitudinal velocity is seen to increase to maximum value of around 6m/s and remains there between 0 and 14 meters of X position. After that the velocity drops to a value of 4.32 m/s at 22 meters of X position. The drop in velocity is as a result of a constraint on roll angle, which is seen to be very close to active region around that same X position as seen in roll angle plot in figure 8.5. Then the velocity is seen to rise again to maximum value of 6m/s after the roll angle reduced due to straight path. There is no drop in longitudinal velocity between 0 and 14 meter of X position although the vehicle is being steered to the left; this is because no constraint is active at that point as can be seen from the graphs.

It is observed from the plots of steering angle and driving torque that when the roll angle constraint is active, it is the torque that is reduced; this ensures the vehicle tracks the Y position as closely as possible. Altering the steering angle would mean moving out of the planned trajectory. This explains why the X position scale on longitudinal velocity plot does not necessarily match that of other plots. The next drop in velocity is to be between 30 and 40 meters of X position as can be logically seen from plot of Y position graph (figure 8.1) and roll angle graph (figure 8.5); this is the X position where roll angle constraint is active again. Therefore in actual sense, the next drop seen on the velocity plot is between 30 and 40 meters of X position.

A plot of driving torque in figure 8.6 shows that the controller observed the constraint on saturation limit. The driving torque reached maximum possible value when the vehicle was accelerating as seen on velocity plot. Similarly on driving torque rate (figure 8.9), the constraint was active as well, this time only once as the driving torque was changing from zero to maximum allowed value of one. From figures 8.7 and 8.8 of vehicle yaw rate and steering rate respectively, it is seen that no constraint on these parameters was active during the simulation.

These results (from figures 8.1 to 8.9) show that the designed NMPC controller using ACADO software package is able to work effectively in tracking and respecting constraints of the nonlinear vehicle model. It is seen that the roll angle constraint is respected throughout the path; this means the vehicle does not rollover at any point. The next thing to be checked is robustness and disturbance rejection capability of the designed controller.

8.2 Robustness of the Controller

After successful performance of the controller during simulation with the exact model it was designed for, it was necessary to check robustness of the controller. Robustness is an important characteristic as it entails controller capability of achieving its goals in times where the controller model is different from the actual (physical) plant model. Model difference comes in due to un-modelled parameters as well as simplification of the model during modelling. This is crucial especially when it comes to implementation; the physical plant is not exactly the same as the mathematical model used during design of the controller.

To test robustness of the proposed controller structure these two steps were done; firstly another trajectory (different from previous one) was designed to be followed by the controller. The purpose of designing another trajectory was to allow high flexibility in manoeuvring of the vehicle before the constraints are activated, i.e. to be able to see clearly the effects of model mismatch. The best would be to see constraints being activated due to model mismatch. Secondly, to incorporate the effect of model mismatch, parameters of the plant (simulation environment external model referred to as process2 in chapter 7) were changed, making it different from the model used in the controller.

Considering the point that for vehicles it is usually the mass that can easily change due to loading, hence changing Centre of Gravity (CoG) height as well as position of CoG; these parameters were changed first in this study for analysis. Firstly, mass was increased by 10%, CoG height was decreased by 5% and CoG position was shifted to the rear by 5%. Secondly, mass was decreased by 10%, CoG height was increased by 5% and CoG position was shifted to the front by 5%. In addition to these two mentioned simulation scenarios of model mismatch effect, other two simulations were performed as well. The first one considered a positive 5% change (model mismatch) in all parameter. The second considered a negative 5% change in all parameter.

The four simulation scenarios for checking robustness of controller to model mismatch are summarised in table 8.1.

Table 8.1: Robustness check simulation scenarios

Simulation	**Change in mass**	**Change in CoG height**	**Change in CoG position**	**Change in other parameters**
Simulation 1	+10%	-5%	-5%	0%
Simulation 2	-10%	+5%	+5%	0%
Simulation 3	+5%	+5%	+5%	+5%
Simulation 4	-5%	-5%	-5%	-5%

Similar to the previous section, the second trajectory designed defines lateral vehicle position in inertia frame (Y) as a function of longitudinal vehicle position in inertia frame (X). Also, the yaw angle reference trajectory was defined as the derivative of Y position with respect to X position. The Y as function of X trajectory was defined as follows:

$$Y_{ref2} = \left(\frac{1.05}{2}\right) * \left(1 + \tanh\left(\left(\frac{5}{25}\right) * \left(X_{ref2} - 18.19\right) - 1.2\right)\right) - \left(\frac{2.7}{2}\right) * \left(1 + \tanh\left(\left(\frac{5}{21.95}\right) * \left(X_{ref2} - 26.46\right)\right)\right). \tag{8.7}$$

The yaw angle reference trajectory was then

$$\Psi_{ref2} = \frac{dY_{ref2}}{dX_{ref2}}. \tag{8.8}$$

Initially the controller was simulated with the original model with its parameters unchanged using this second trajectory (equations 8.7 and 8.8). The aim of doing this was to come up with a benchmark (reference) for ease of comparison of results when model changes are introduced later. The model with unchanged parameters has been named *Model-1*, while the changed models representing simulation 1, simulation 2, simulation 3 and simulation 4 have been named *Model-2, Model-3, Model-4* and *Model-5* respectively. Simulations with Model-1 are presented first, then those with changed models follow in the same order they appear in table 8.1.

As previously done, all the results have been plotted against X position coordinate of inertia frame. For simulations with Model-1, the reference YX trajectory and actual YX trajectory followed by the controller have been plotted on the same graph, similarly the reference Ψ and actual Ψ have also been plotted on the same graph. After these graphs, then roll angle, steering angle and longitudinal velocity are plotted as well.

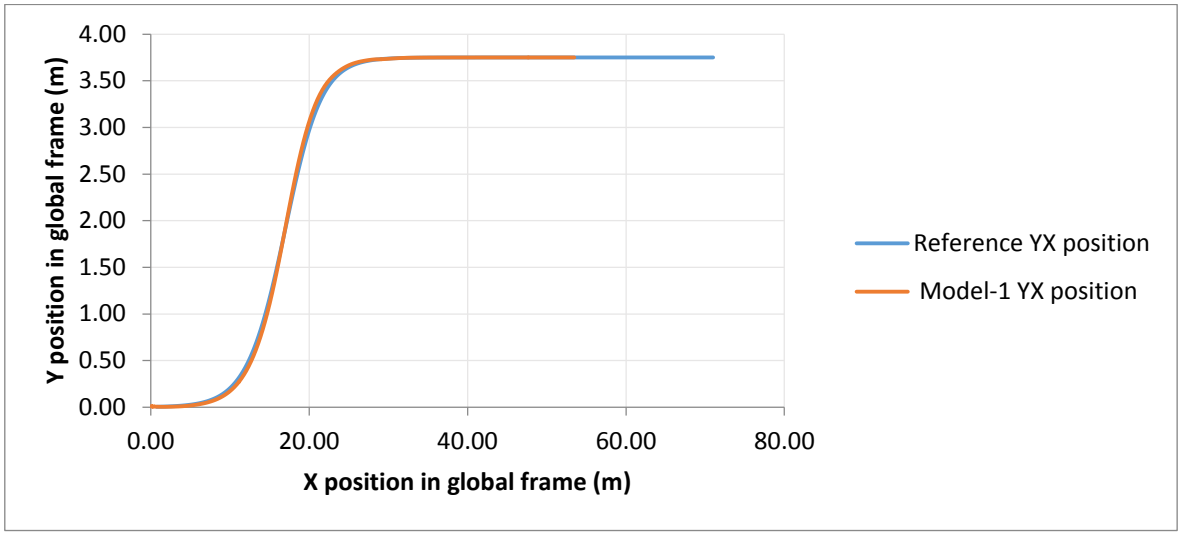

Figure 8.10: Showing reference YX position and Model-1 YX position trajectories plotted against X position in global coordinate frame, the reference is tracked with less than 0.05 meters error.

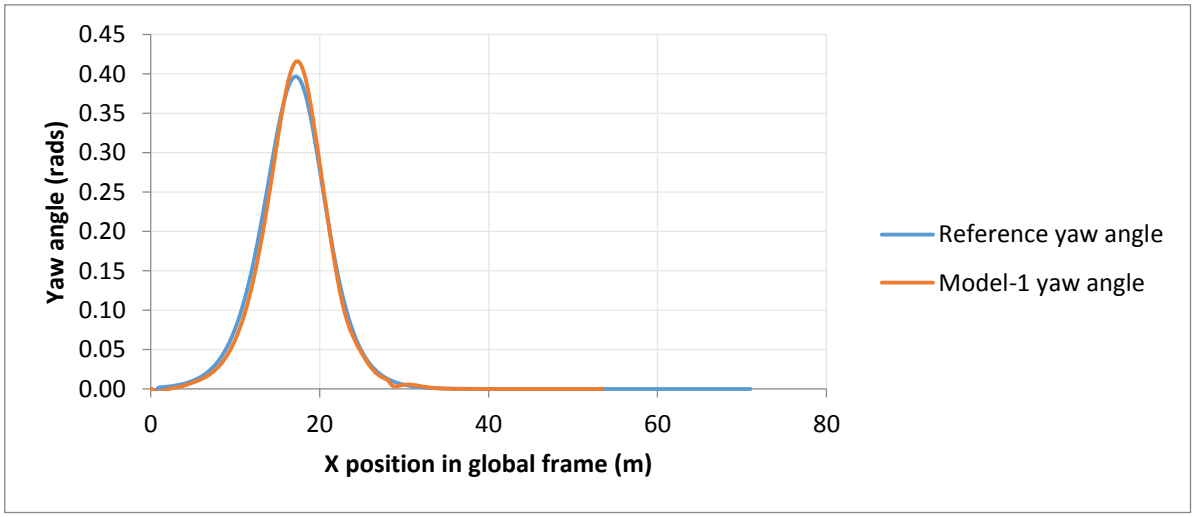

Figure 8.11: Showing reference yaw angle and Model-1 yaw angle trajectories plotted against X position in global coordinate frame, the reference is tracked with maximum error of 0.02 radians at 18.1 meters of X position.

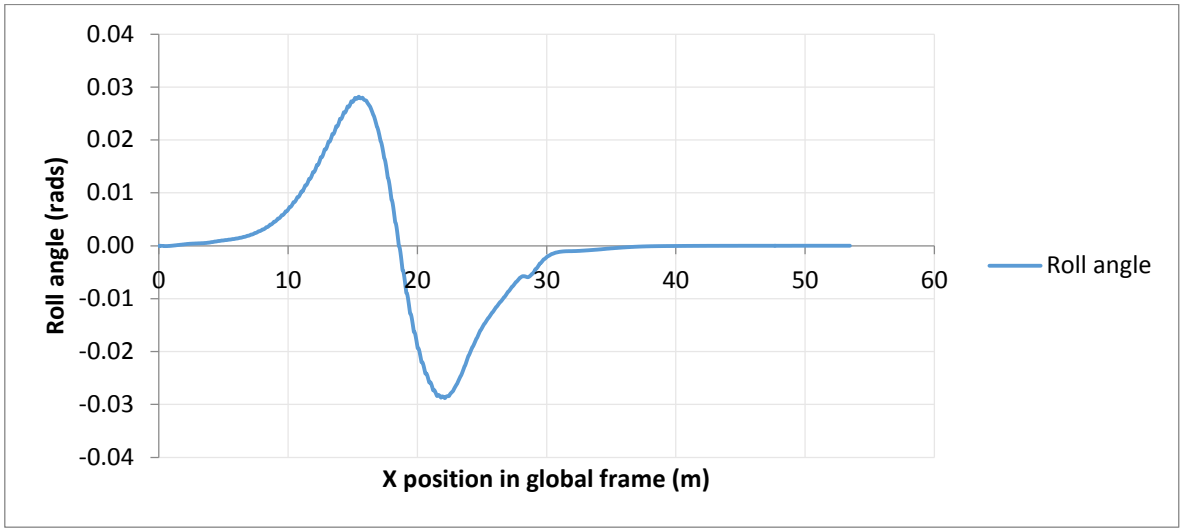

Figure 8.12: Showing Model-1 vehicle roll angle plotted against X position, roll angle is seen to be within safe zone of operation (roll angle constraint is not activated).

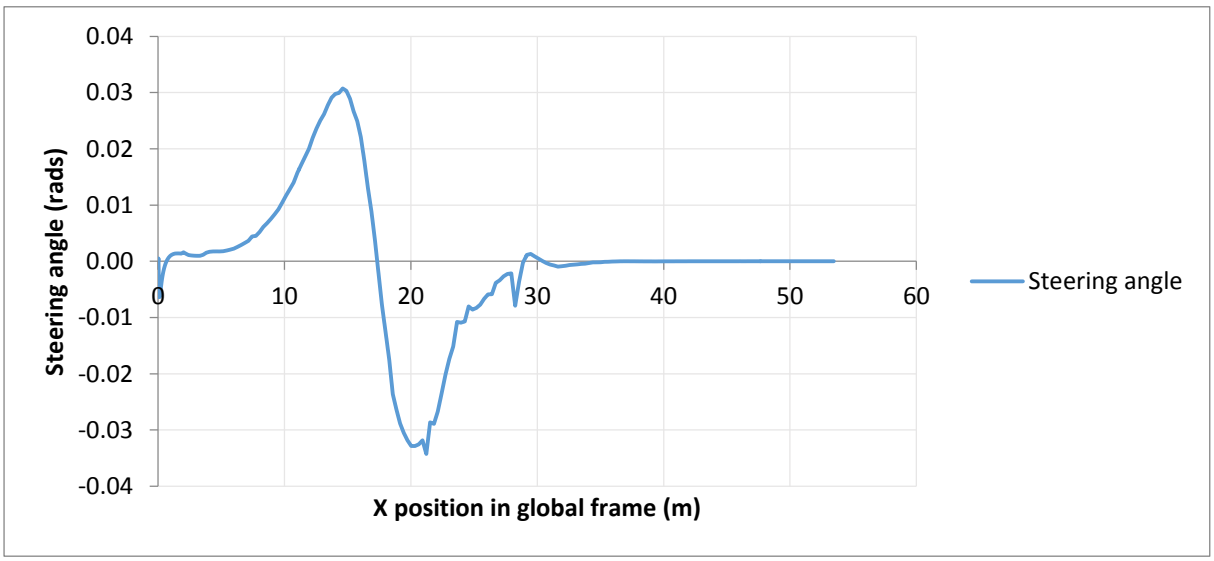

Figure 8.13: Showing Model-1 vehicle steering angle plotted against X position, steering angle is seen changing with changing reference YX trajectory to best follow it (limit is not reached).

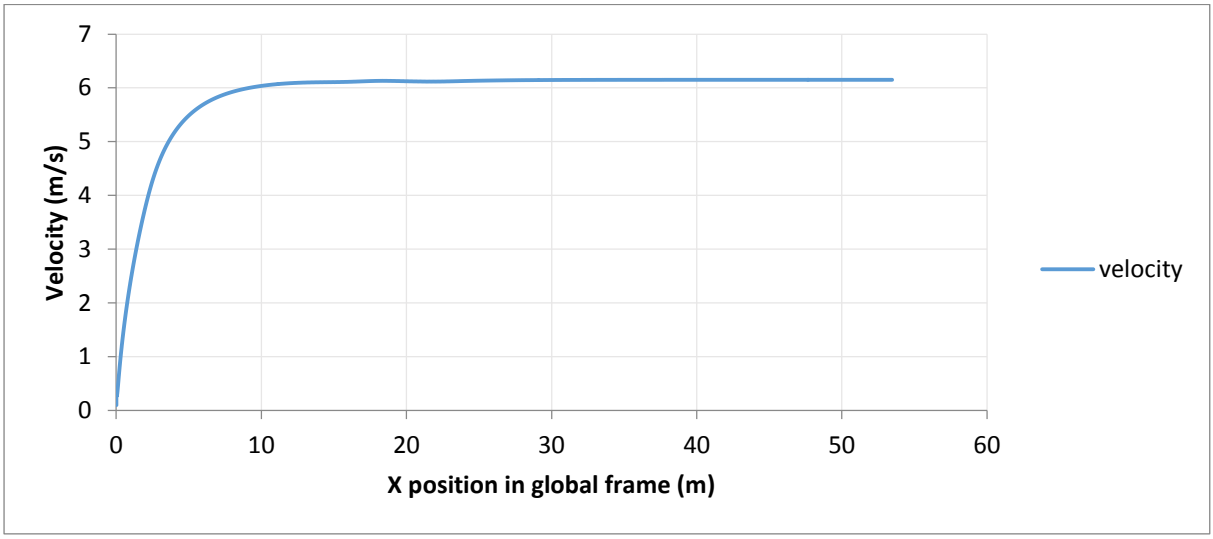

Figure 8.14: Showing Model-1 vehicle longitudinal velocity plotted against X position, velocity is seen increasing to maximum value and remain there because no constraint is active so far.

From figures 8.10, 8.11 and 8.14, it is observed that the vehicle is able to track very well the reference YX and yaw angle trajectories at a uniform maximum velocity. YX trajectory tracking error is less than 0.05 meters which is less than the maximum allowed of which is 0.1 meters. The velocity does not change along the path of travel because no constraint is active as can be seen from the roll angle plot (figure 8.12) and steering angle plot (figure 8.13). No constraint was activated on yaw rate, roll rate, steering rate and driving torque rate as well. There graphs have not been plotted for brevity purposes.

For simulations with Model-2, Model-3, Model-4 and Model-5, the reference YX trajectory and actual YX trajectory followed by the controller have been plotted on the same graph for each model. Similarly the reference Ψ and actual Ψ have also been plotted on the same graph. After these graphs, then roll angle of each is plotted on the same graph with the roll angle of Model-1, in a similar way steering angle and longitudinal velocity are plotted. The idea of plotting these graphs together is to see clearly the effect of model mismatch on controller performance. The graphs showing yaw rate, roll rate, steering rate and driving torque rate are not plotted for clarity purposes; only for Model-2 are plotted in appendix A of this report.

- **Simulation 1 (Model-2)**
  (10% increase in mass, 5% decrease in CoG height, 5% decrease in CoG position from rear end and 0% change in all other parameters.)

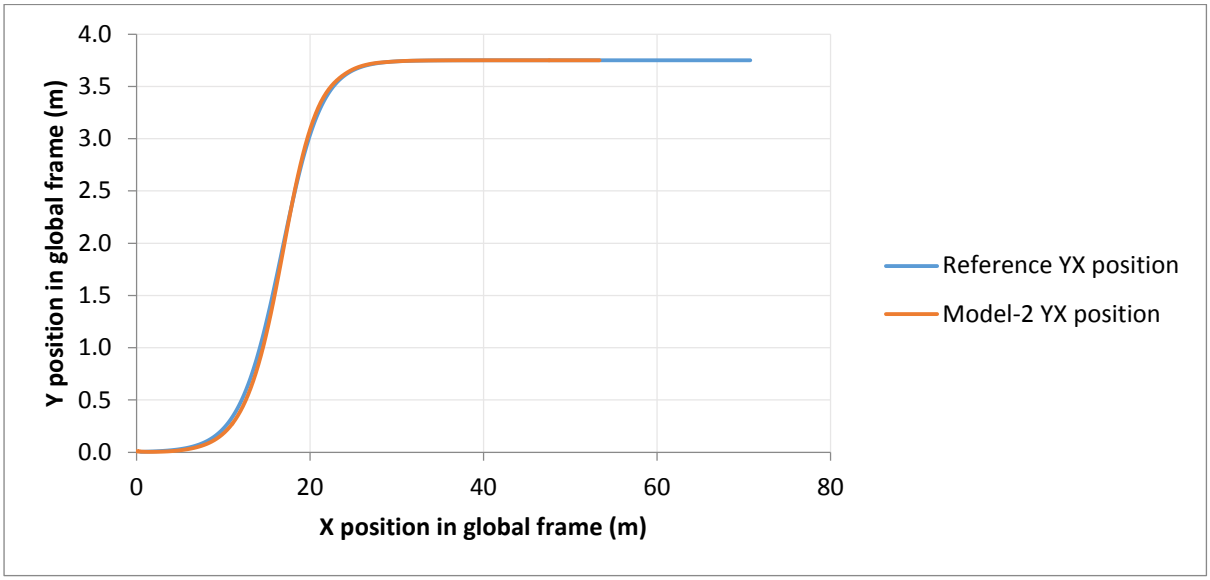

Figure 8.15: Showing reference YX position and Model-2 YX position trajectories plotted against X position in global coordinate frame, the reference is tracked with a maximum error of 0.05 meters.

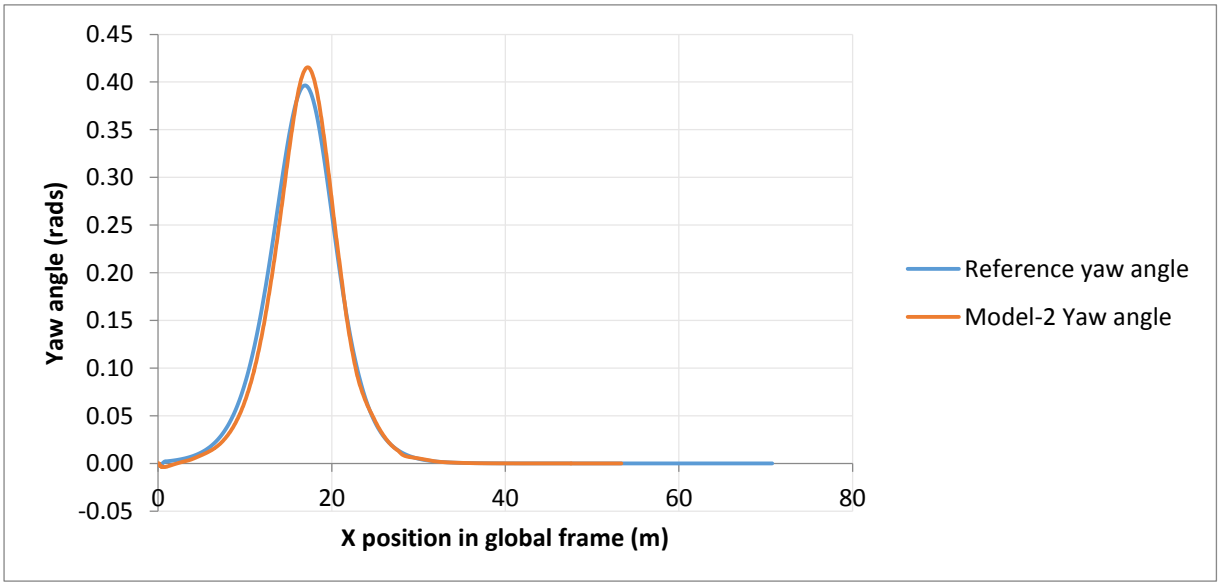

Figure 8.16: Showing reference yaw angle and Model-2 yaw angle trajectories plotted against X position in global coordinate frame, the reference is tracked with slightly larger error (0.03 radians) as compared to Model-1 graph in figure 8.11, due to model mismatch.

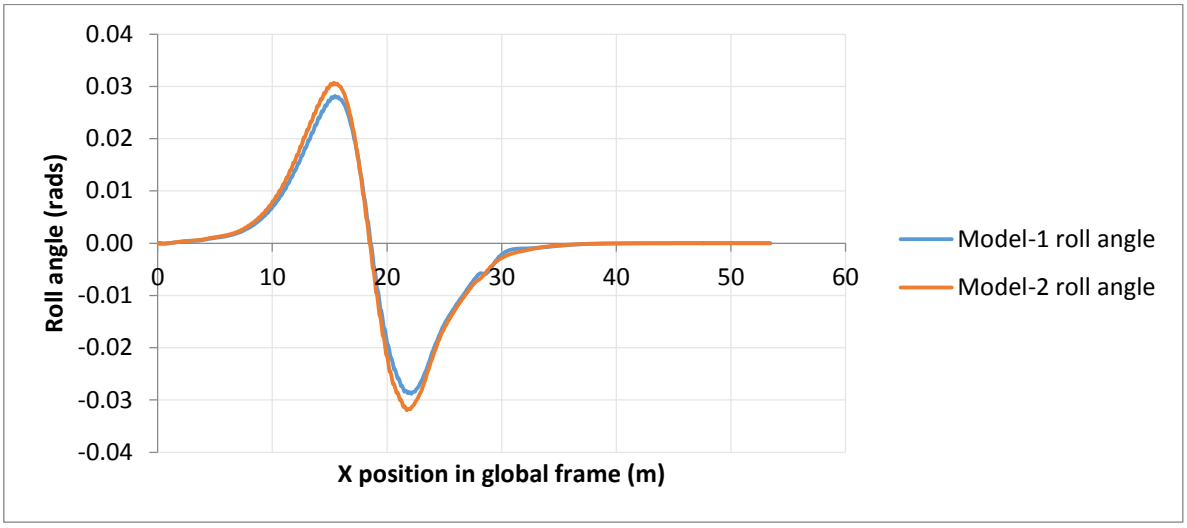

Figure 8.17: Showing Model-2 and Model-1 vehicle roll angles plotted against X position, difference in peak values due to model mismatch is observed in the graphs. Model-2 has higher peak values (0.03 and -0.031 radians), but roll angle limit is not reached.

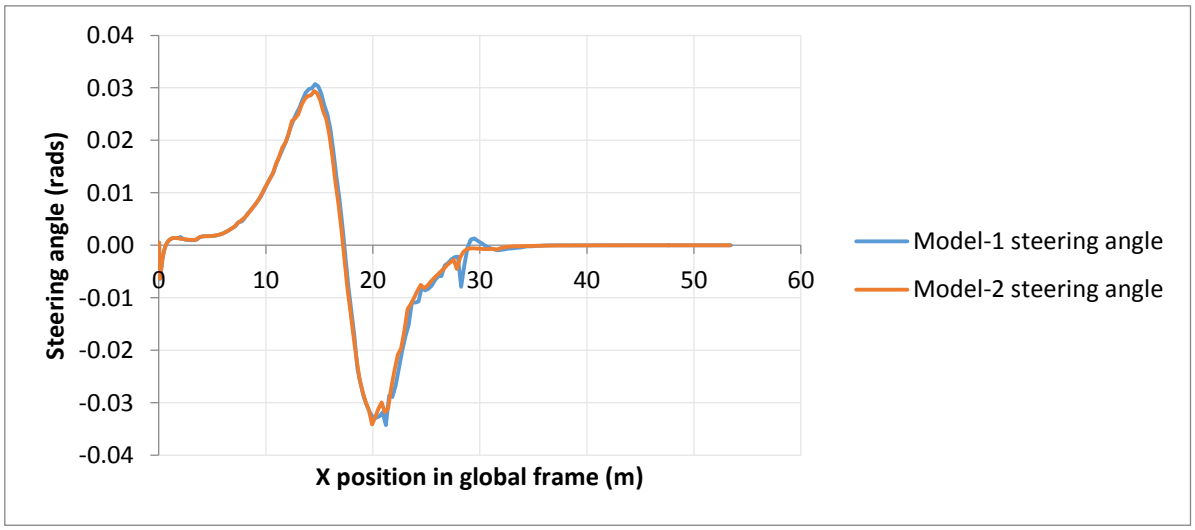

Figure 8.18: Showing Model-2 and Model-1 vehicle steering angles plotted against X position. Very small difference (maximum 0.005 radians) is seen because they are both tracking same trajectory and the controller needs to follow it.

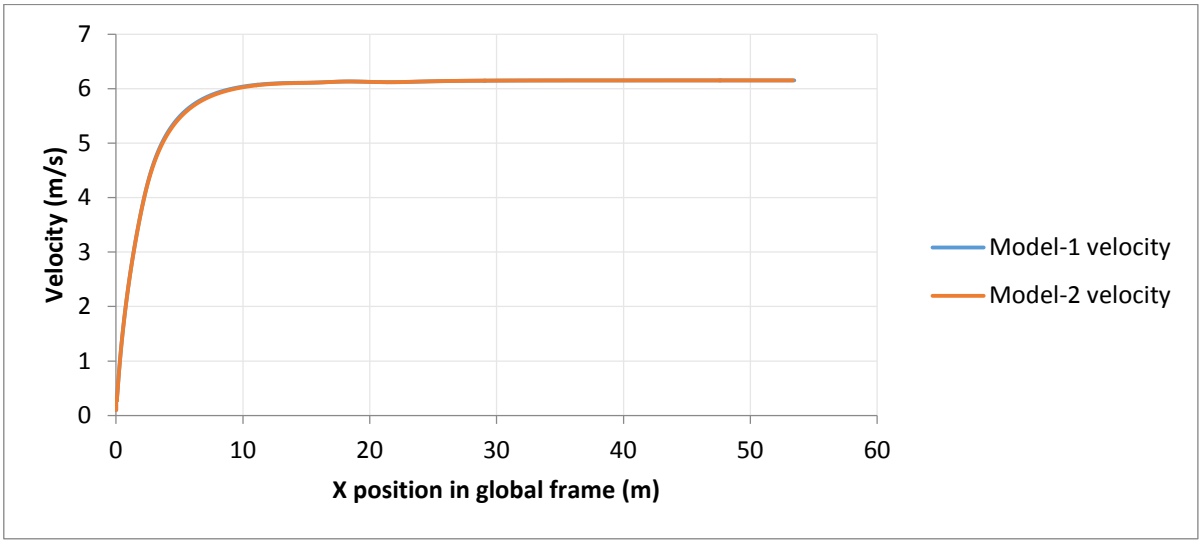

Figure 8.19: Showing Model-2 and Model-1 vehicle longitudinal velocities plotted against X position, the velocities are almost exactly the same because no constraint is active for the whole path.

From Model-2 simulation graphs, it is observed that vehicle is still able to track very well the reference YX position and yaw angle trajectories as seen in figures 8.15 and 8.16 respectively. The vehicle did not go outside the defined trajectory. The effect of the introduced model mismatch is seen on roll angle and steering angle graphs in figures 8.17 and 8.18 respectively. It is seen that the magnitude of the roll angle for Model-2 is slightly higher than that of Model-1 at their peak values. Very small differences (maximum of 0.005 radians) are seen in the steering angles. In both roll angle and steering angle graphs, no constraint is activated hence the vehicle still moves at the maximum possible velocity as seen in figure 8.19.

- **Simulation 2 (Model-3)**
  (10% decrease in mass, 5% increase in CoG height, 5% increase in CoG position from rear end and 0% change in all other parameters.)

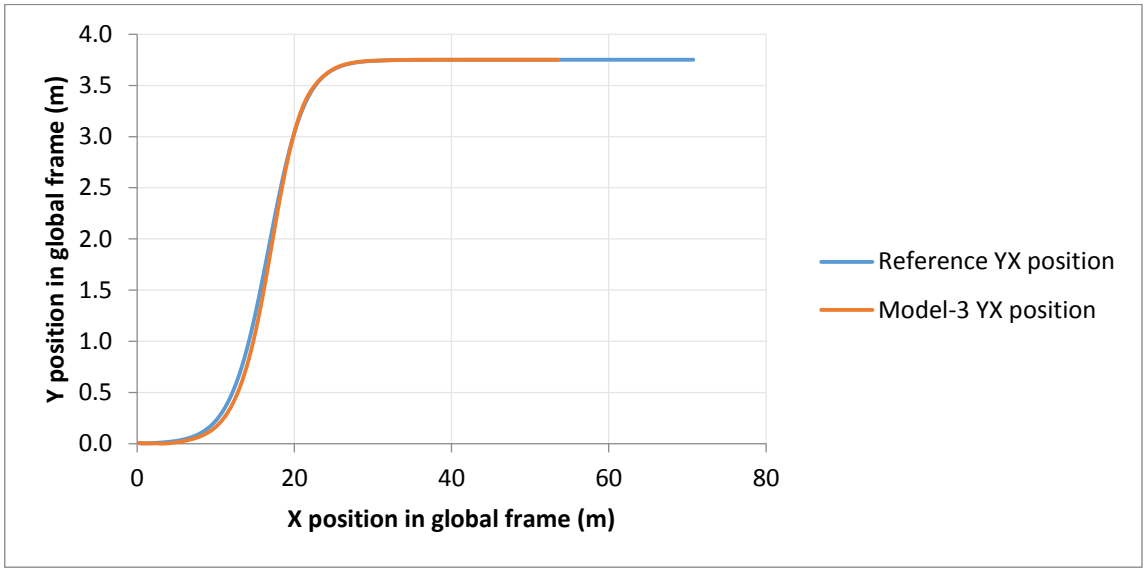

Figure 8.20: Showing reference YX position and Model-3 YX position trajectories plotted against X position in global coordinate frame, the reference is tracked with a maximum error of 0.1 meters.

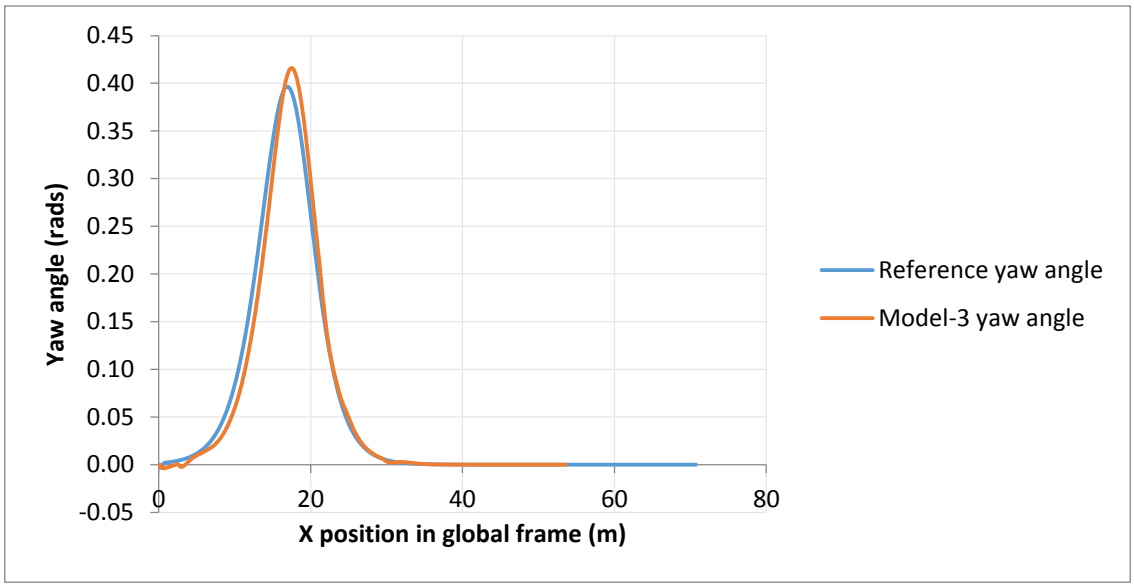

Figure 8.21: Showing reference yaw angle and Model-3 yaw angle trajectories plotted against X position in global coordinate frame, the reference is tracked with slightly larger error as compared to Model-1 graph in figure 8.11, due to model mismatch. Maximum error is 0.03 radians.

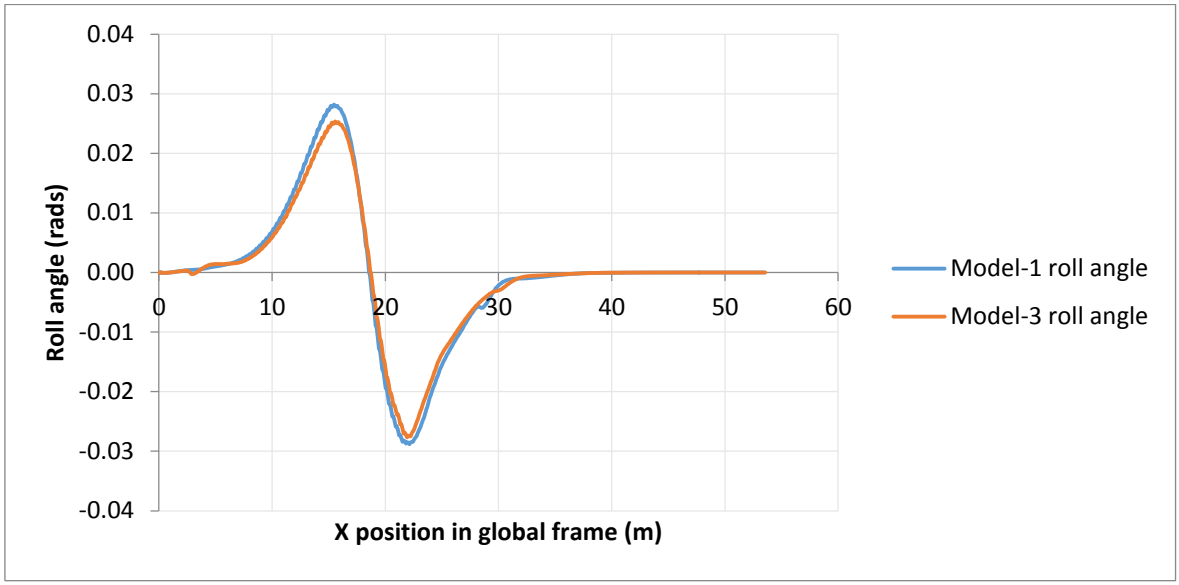

Figure 8.22: Showing Model-3 and Model-1 vehicle roll angles plotted against X position, difference in peak values due to model mismatch is observed in the graphs. Model-1 has peak values of 0.028 and -0.028 radians. Model-3 has lower peak values of 0.025 and -0.027 radians.

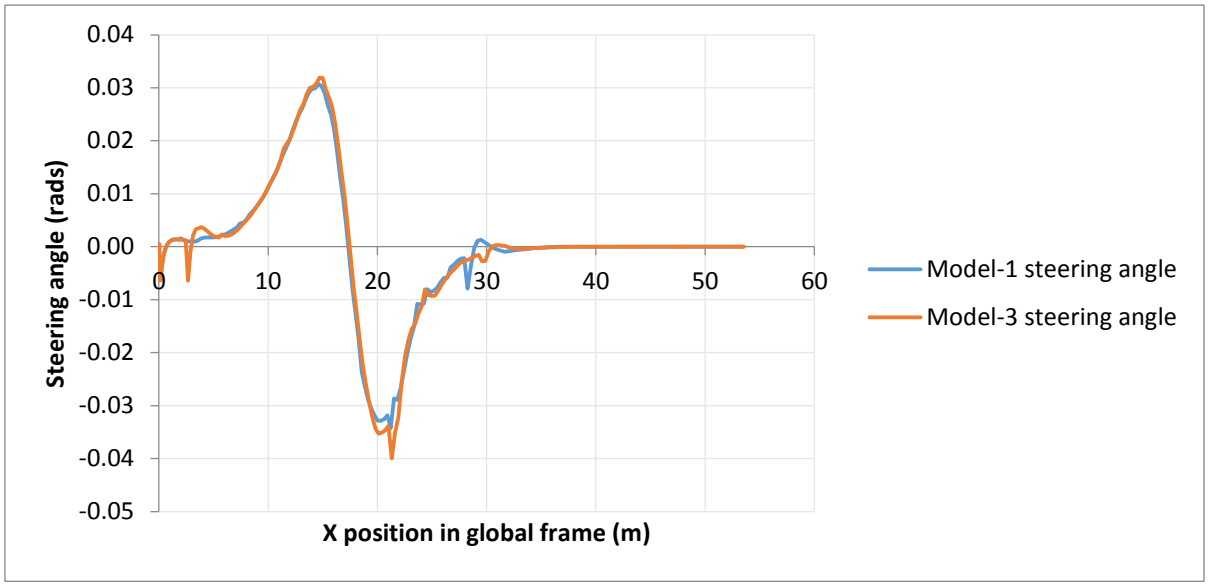

Figure 8.23: Showing Model-3 and Model-1 vehicle steering angles plotted against X position. Very small difference (maximum of 0.007 radians) is seen because they are both tracking same trajectory and the controller needs to follow it.

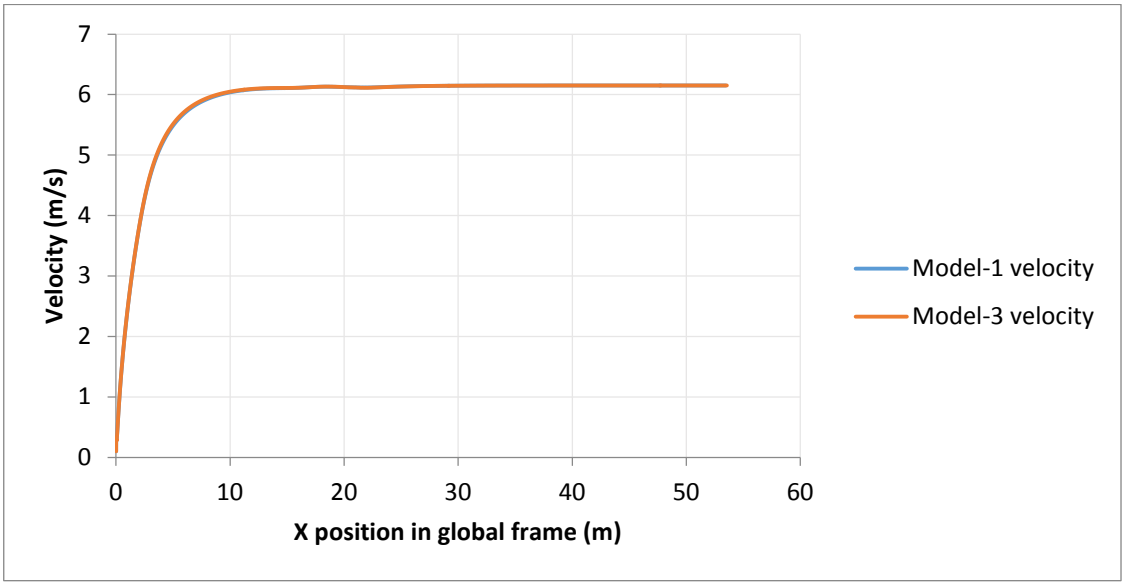

Figure 8.24: Showing Model-3 and Model-1 vehicle longitudinal velocities plotted against X position, the velocities are almost exactly the same because no constraint is active for the whole path.

From Model-3 simulation graphs, it is observed that vehicle is still able to track very well the reference YX position and yaw angle trajectories as seen in figures 8.20 and 8.21 respectively. The effect of the introduced model mismatch is seen on roll angle and steering angle graphs in figures 8.22 and 8.23 respectively. It is seen that the magnitude of the roll angle for Model-3 is slightly lower than that of Model-1 at their peak values. Very small differences (0.007 radians) are seen in the steering angles. In both roll angle and steering angle graphs, no constraint is activated hence the vehicle still moves at the maximum possible velocity as seen in figure 8.24.

- **Simulation 3 (Model-4)**

  (5% increase in all parameters.)

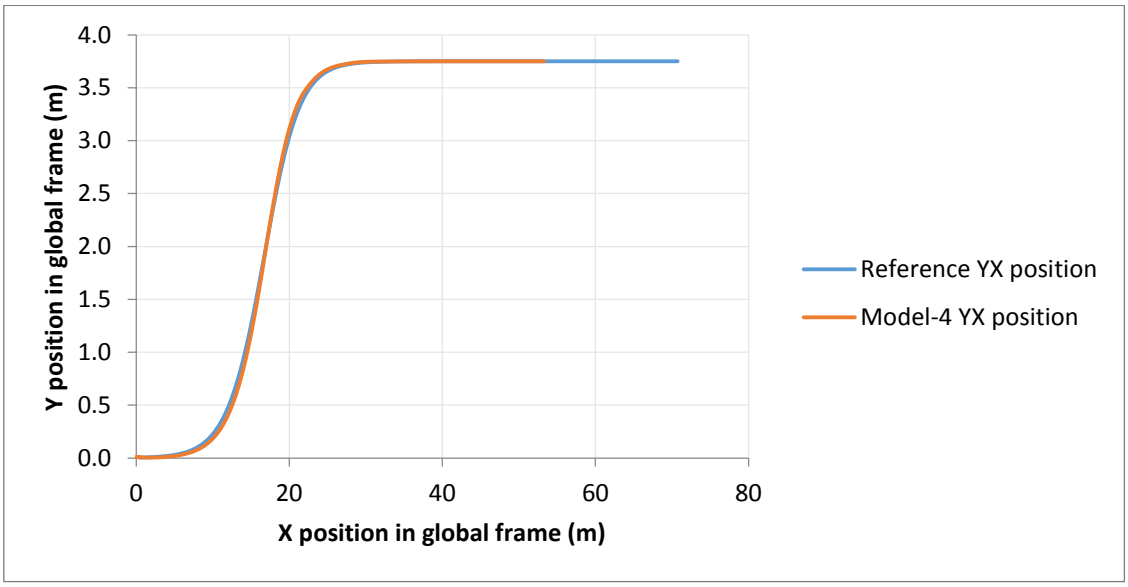

Figure 8.25: Showing reference YX position and Model-4 YX position trajectories plotted against X position in global coordinate frame, the reference is tracked with maximum error of 0.06 meters.

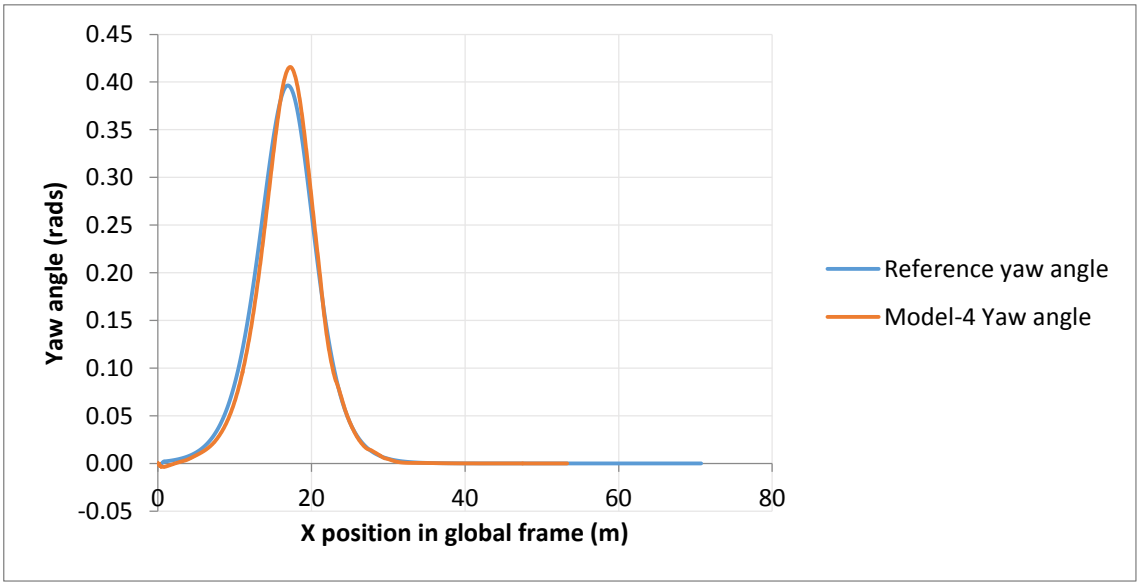

Figure 8.26: Showing reference yaw angle and Model-4 yaw angle trajectories plotted against X position in global coordinate frame, the reference is tracked with slightly larger error (maximum of 0.03 radians at peak value) as compared to Model-1 graph in figure 8.11, due to model mismatch.

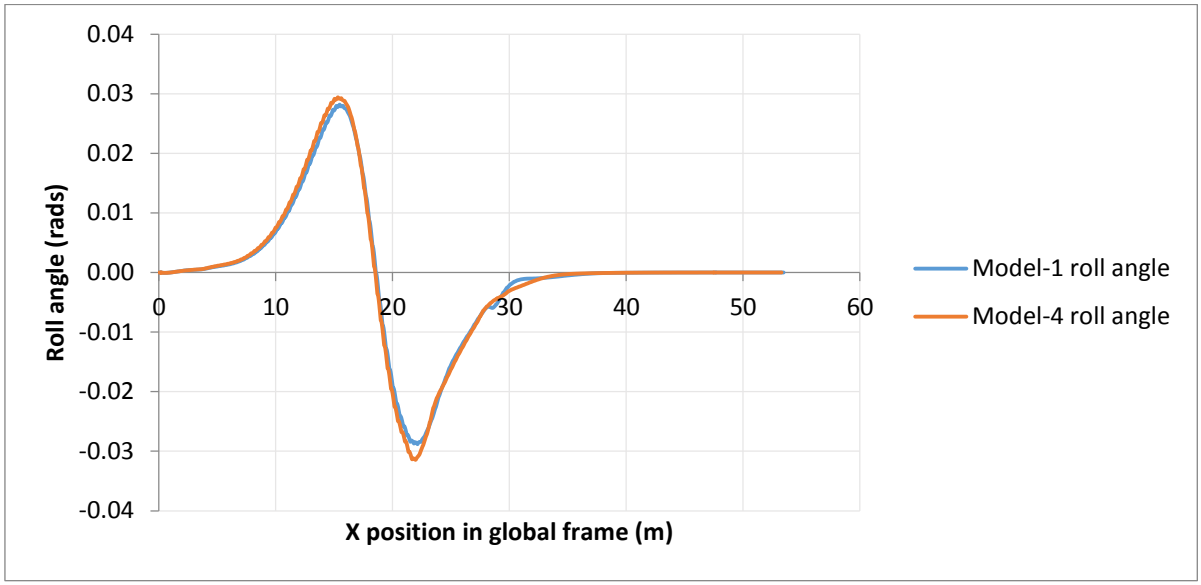

Figure 8.27: Showing Model-4 and Model-1 vehicle roll angles plotted against X position, difference in peak values due to model mismatch is observed in the graphs. Model-4 has higher peak values of 0.029 and -0.031 radians. Model-1 has peak values of 0.028 and -0.028 radians. Roll angle limit is not reached.

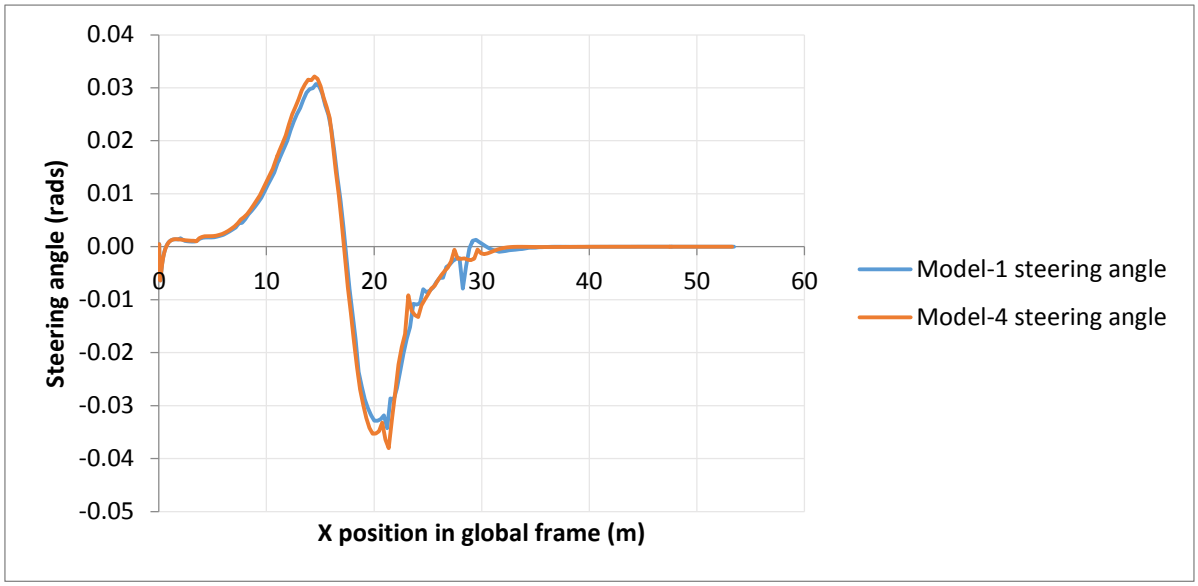

Figure 8.28: Showing Model-4 and Model-1 vehicle steering angles plotted against X position. Very small difference (maximum of 0.005 radians) is seen because they are both tracking same trajectory and the controller needs to follow it.

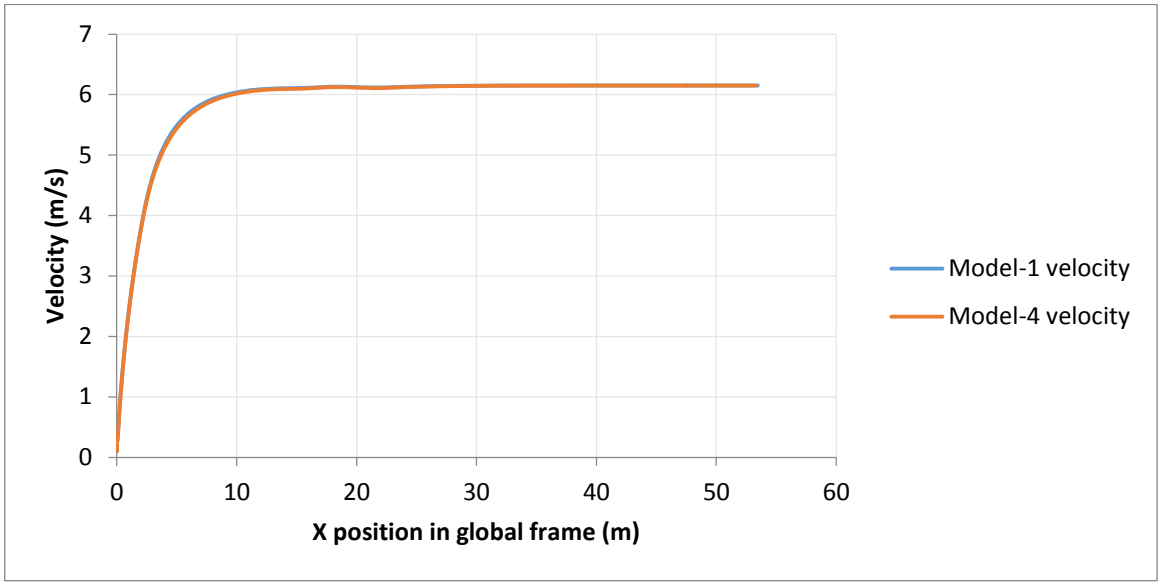

Figure 8.29: Showing Model-4 and Model-1 vehicle longitudinal velocities plotted against X position, the velocities are same because no constraint is active for the whole path.

From Model-4 simulation graphs, it is observed that vehicle is still able to track very well the reference YX position and yaw angle trajectories as seen in figures 8.25 and 8.26 respectively. The effect of the introduced model mismatch is seen on roll angle and steering angle graphs in figures 8.27 and 8.28 respectively. It is seen that the magnitude of the roll angle for Model-4 is slightly higher than that of Model-1 at their peak values. Model-4 roll angle is seen to be higher at negative peak value than at positive peak value. Very small differences (maximum of 0.005 radians) are seen in the steering angles. In both roll angle and steering angle graphs, no constraint is activated hence the vehicle still moves at the maximum possible velocity as seen in figure 8.29.

- **Simulation 4 (Model-5)**
  (5% decrease in all parameters.)

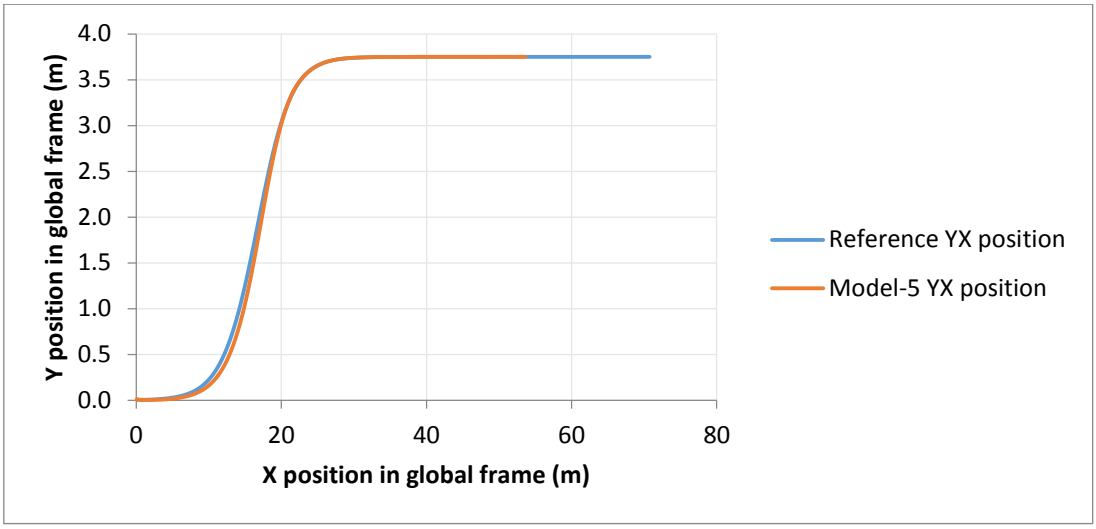

Figure 8.30: Showing reference YX position and Model-5 YX position trajectories plotted against X position in global coordinate frame, the reference is tracked with maximum error of 0.1 meters.

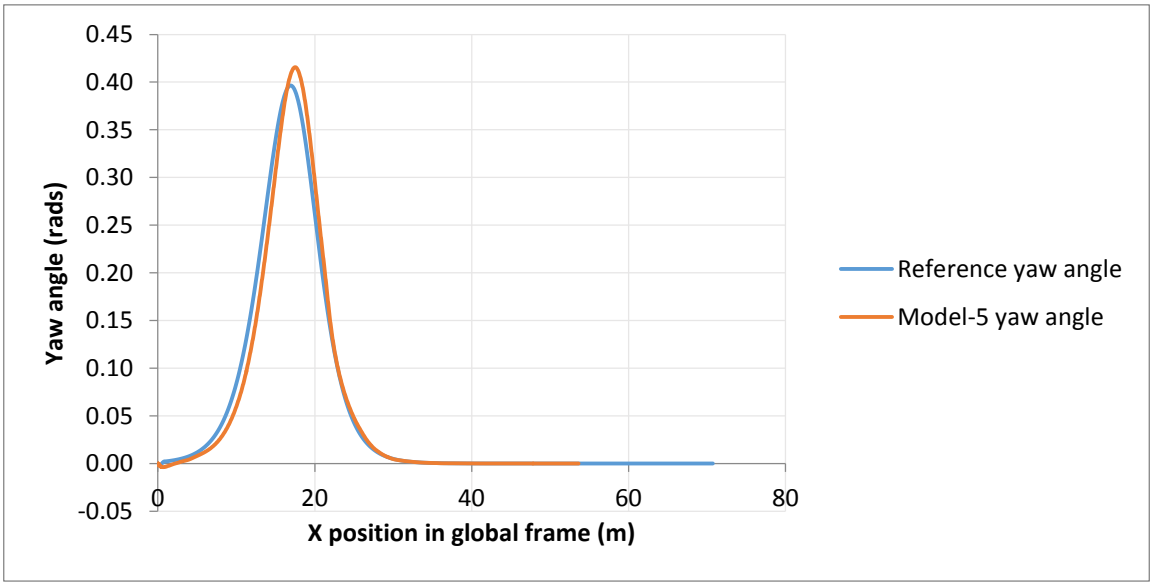

Figure 8.31: Showing reference yaw angle and Model-5 yaw angle trajectories plotted against X position in global coordinate frame, the reference is tracked with slightly larger error (maximum of 0.03 radians at peak value) as compared to Model-1 graph in figure 8.11, due to model mismatch.

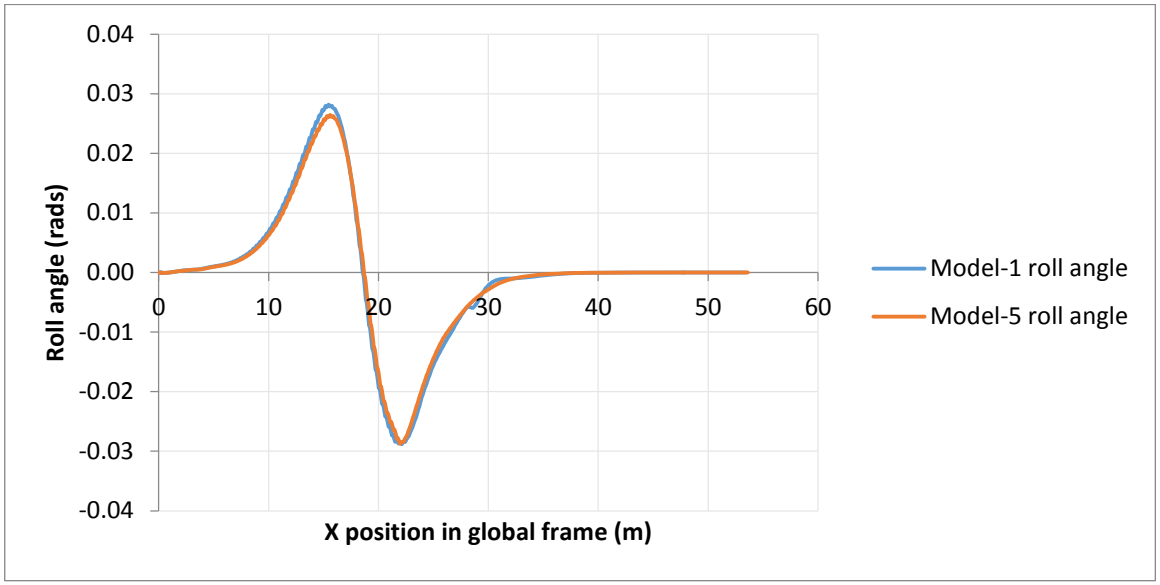

Figure 8.32: Showing Model-5 and Model-1 vehicle roll angles plotted against X position, difference in peak values due to model mismatch is observed in the graphs. Model-5 has peak values of 0.026 and -0.028. Model-1 has peak values of 0.028 and -0.028 radians. Roll angle limit is not reached.

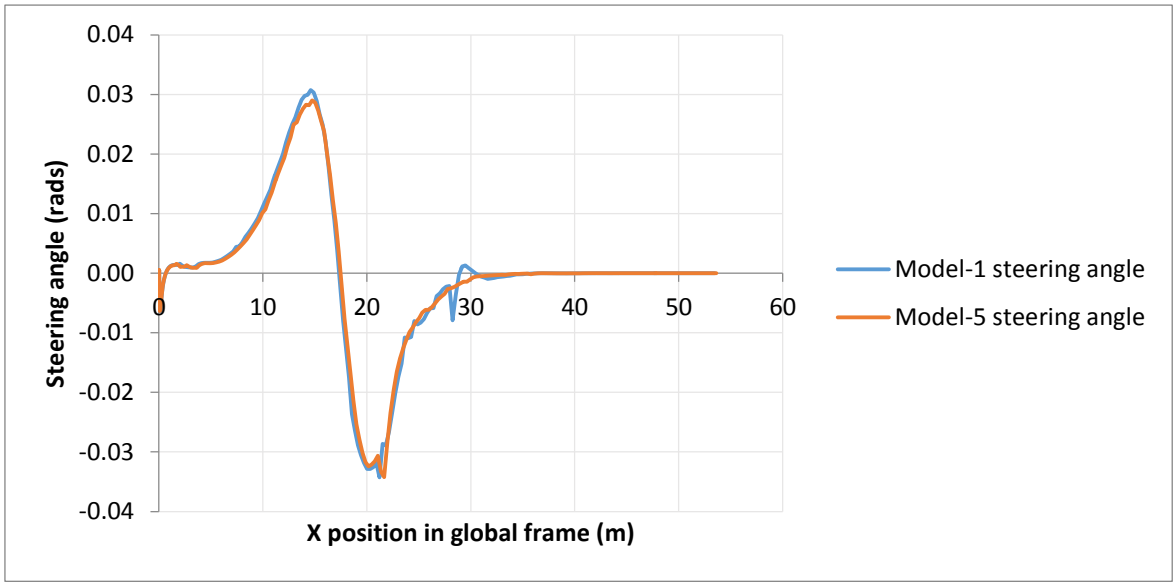

Figure 8.33: Showing Model-5 and Model-1 vehicle steering angles plotted against X position. Very small difference (maximum of 0.005 radians) is seen because they are both tracking same trajectory and the controller needs to follow it.

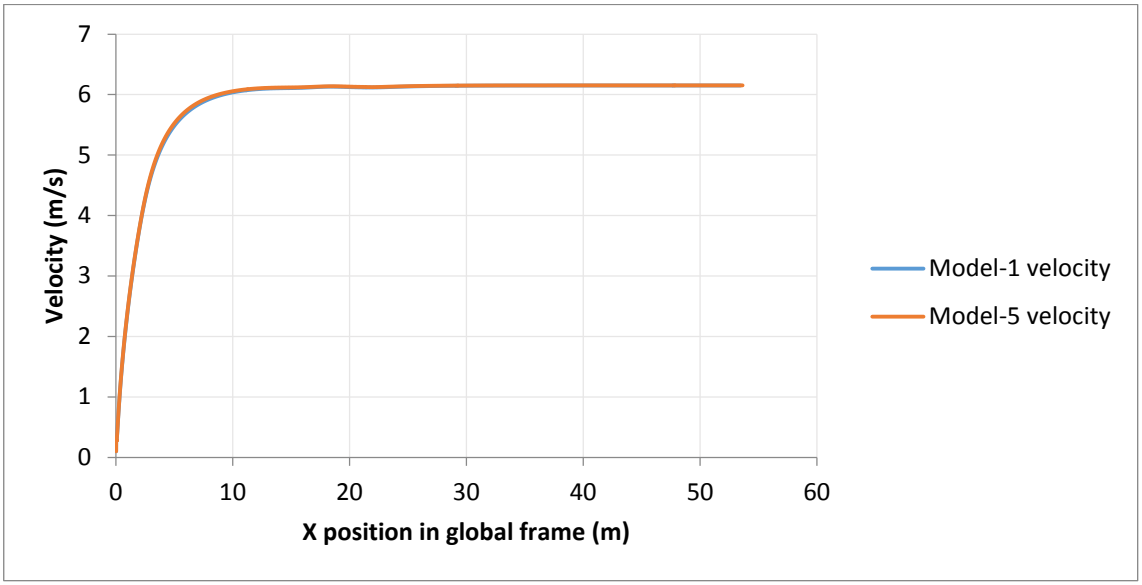

Figure 8.34: Showing Model-5 and Model-1 vehicle longitudinal velocities plotted against X position, the velocities same because no constraint is active for the whole path.

From Model-5 simulation graphs, it is observed that vehicle is still able to track very well the reference YX position and yaw angle trajectories as seen in figures 8.30 and 8.31 respectively. The effect of the introduced model mismatch is seen on roll angle and steering angle graphs in figures 8.32 and 8.33 respectively. It is seen that the magnitude of the roll angle for Model-5 is slightly lower than that of Model-1 at positive peak value. Very small differences (0.005 radians) are seen in the steering angles. In both roll angle and steering angle graphs, no constraint is activated hence the vehicle still moves at the maximum possible velocity as seen in figure 8.34.

From the results, based on controller robustness check simulation scenarios shown in table 8.1, it is concluded that the designed NMPC controller is robust. It has been observed through the plotted results under simulation 1 through simulation 4, that the introduced model mismatch have small effect on the performance of the controller. In general, the controller is able to track the reference trajectories in the situations of model mismatch.

8.3 Disturbance Rejection of the Controller

Another important characteristic tested for the designed controller is its ability to reject disturbances. In this study, disturbance in roll angle was looked into and analysed. Roll angle disturbance of the vehicle can be as a result of the vehicle running over an obstacle, a trench or travelling on the road that is not perfectly flat (horizontal) (more information is in the literature review). All these causes of disturbance can be at any point of the trajectory to be followed, that is whether cornering or not.

To simulate the behaviour of the controller in the presence of any of the mentioned disturbances and at any point, pulse disturbances in roll angle were injected into the plant model. Recall that in this study more emphasis has been on preventing vehicle rollover and that rollover occurs when roll angle exceeds 0.05 radians (for this study). This limit on roll angle was chosen in stability analysis chapter.

In regards to explanation in the previous paragraph, two positive pulse disturbances were selected to analyse the performance of the controller. Positive pulses of 0.02 radians and 0.05 radians were chosen as disturbance inputs (negative pulses produced same but inverted results). The pulse of magnitude 0.02 radians represented small disturbances in relation to roll angle limit of 0.05 radians. The pulse of magnitude 0.05 radians represented the logically biggest disturbance that could enter the vehicle system in this study. A disturbance of more than 0.05 radians would actually mean the vehicle has already rolled over. This is because roll angle limit was set to 0.05 radians in this study.

For uniformity and continuity in presentation of results, the models used in this section (which consider disturbances) have been named *Model-6* and *Model-7.* The simulations have been named simulation 5 and simulation 6 for step disturbances of 0.02 radians and 0.05 radians respectively. The combinations are summarised in table 8.2.

Table 8.2: Disturbance rejection simulations

Simulation	**Model**	**Disturbance magnitude**
Simulation 5	Model-6	0.02 radians
Simulation 6	Model-7	0.05 radians

Similar to previous section, for simulations with Model-6 and Model-7, the reference YX trajectory and actual YX trajectory followed by the controller have been plotted on the same graph, similarly the reference Ψ and actual Ψ have also been plotted on the same graph. After these graphs, then roll angles of Model-5 and Model-6 are plotted on the same graph with the roll angle of Model-1 (*Model-1* was described in previous section), in a similar way steering angles and longitudinal velocities are plotted. The idea of plotting these graphs together is to clearly see the effect of input disturbances on controller performance. The graphs showing yaw rate, roll rate, steering rate and driving torque rate are not plotted in this section for brevity and clarity purposes; only Model-6 graphs (yaw rate, roll rate, steering rate and driving torque rate) are plotted in appendix A of this report.

- **Simulation 5 (Model-6)**
  (0.02 radians disturbance)

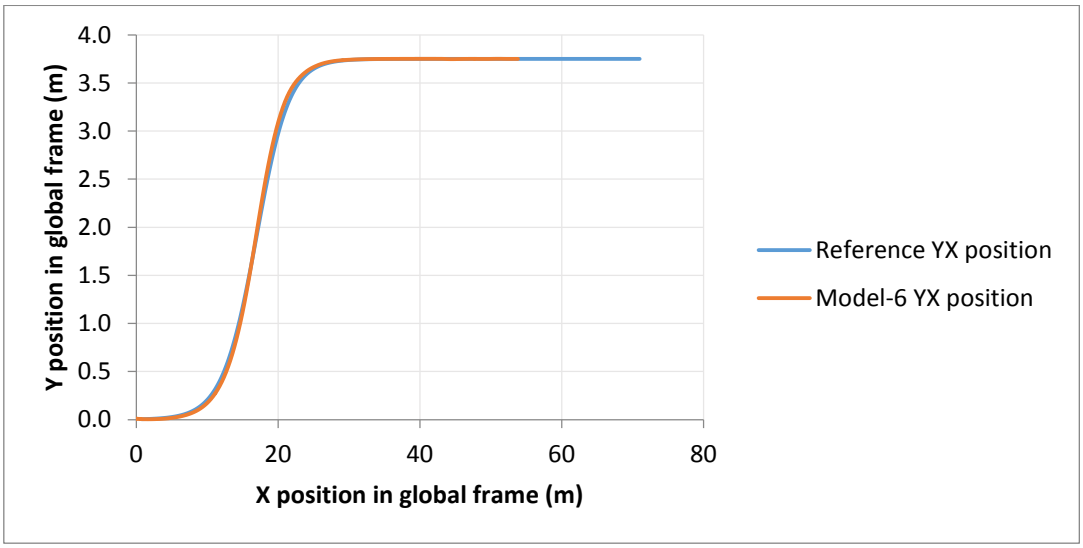

Figure 8.35: Showing reference YX position and Model-6 YX position trajectories plotted against X position in global coordinate frame. Reference is seen being tracked with maximum error of 0.05 meters.

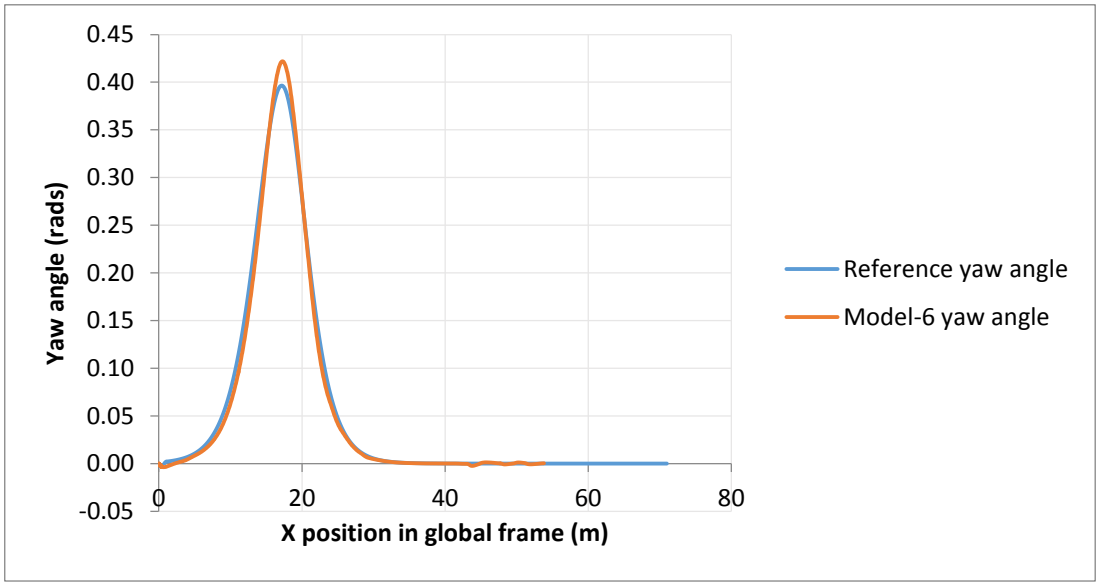

Figure 8.36: Showing reference yaw angle and model-6 yaw angle trajectories plotted against X position in global coordinate frame. Reference is tracked with maximum error of 0.03 radians at peak yaw angle.

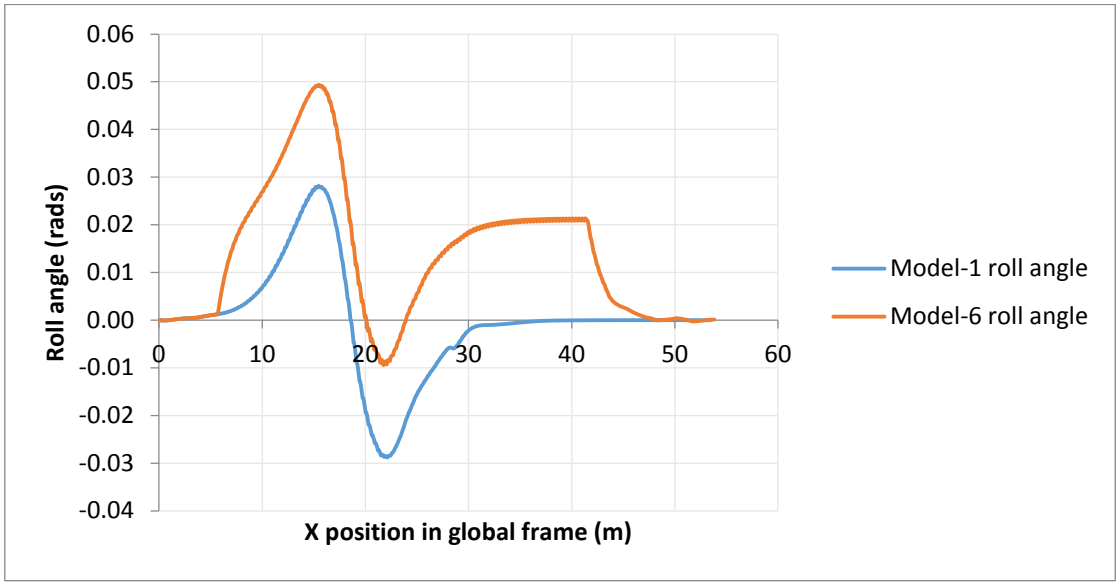

Figure 8.37: Showing Model-6 and Model-1 vehicle roll angles plotted against X position. Model-6 roll angle is seen shifted upwards between 5 and 42 meters of X position due to injected positive pulse disturbance.

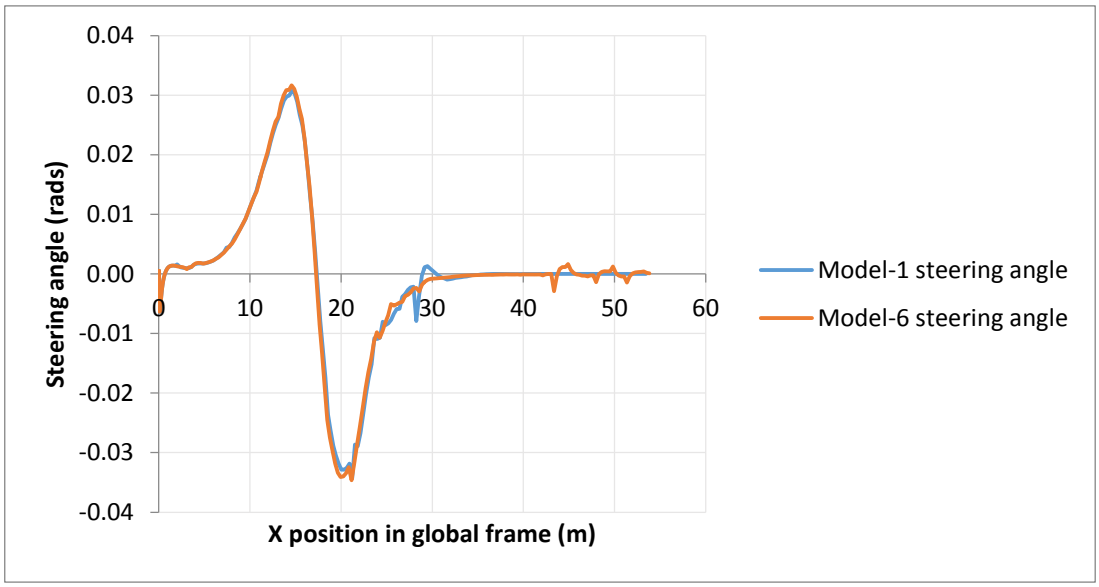

Figure 8.38: Showing Model-6 and Model-1 vehicle steering angles plotted against X position. Very small difference (maximum of 0.005 radians) in steering angle is seen because they are tracking same trajectory.

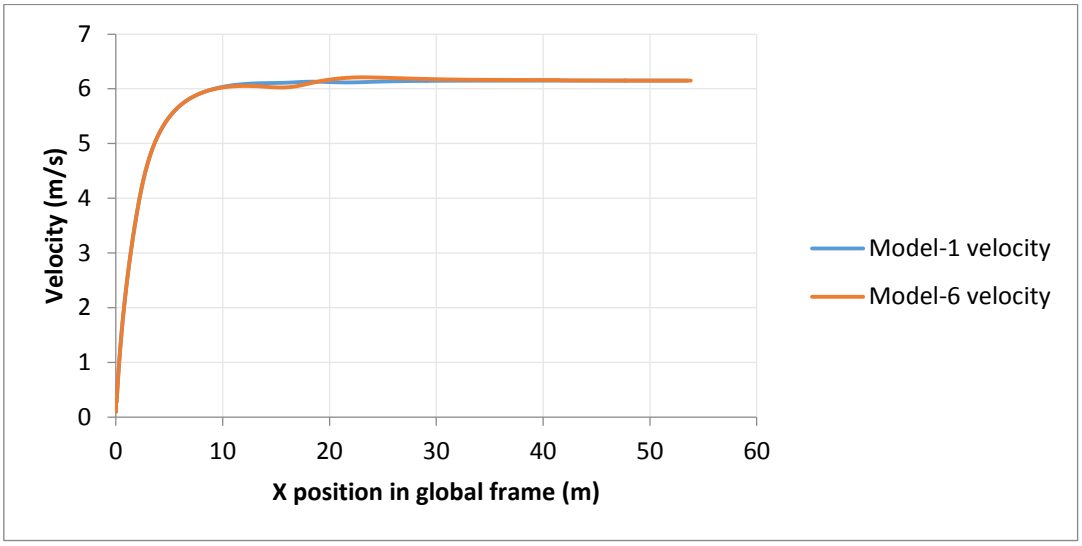

Figure 8.39: Showing Model-6 and Model-1 vehicle longitudinal velocities plotted against X position. Difference in velocity is observed between 10 and 20 meters of X position, this is due to injected roll angle disturbance which almost activated roll angle constraint between those points as can be seen in figure 8.37.

After introducing a 0.02 radians step disturbance in roll angle (figure 8.37), it is observed that the vehicle is still able to track very well the reference YX and yaw angle trajectories as seen in figures 8.35 and 8.36 respectively. YX trajectory tracking error is 0.05 meters which is less than the allowed maximum value of 0.1 meters. The roll angle disturbance resulted in the roll angle of the vehicle being shifted upwards by almost 0.02 radians as can be seen in figure 8.37. In reference to model-1 (same figure 8.37), it is observed that the introduced roll disturbance pushes the vehicle's roll angle very close to the defined constraint boundary during first cornering. It is seen in figure 8.39 that this closeness to the constraint boundary resulted in velocity being reduced to avoid violating the constraint (preventing vehicle rollover). After roll angle decreased, the velocity is seen picking up to maximum possible value again.

- **Simulation 6 (Model-7)**
  (0.05 radians disturbance)

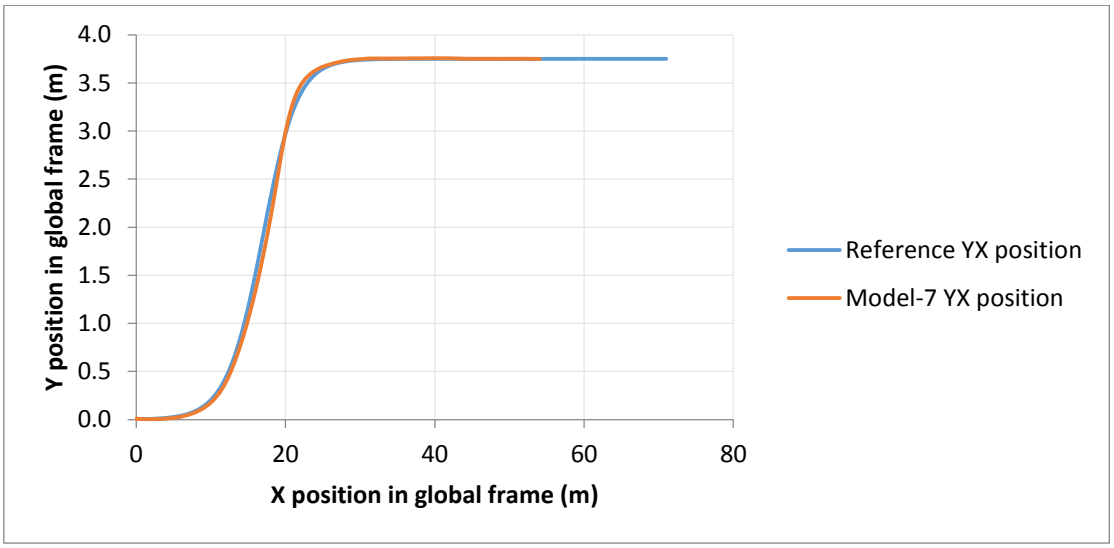

Figure 8.40: Showing reference YX position and Model-7 YX position trajectories plotted against X position in global coordinate frame. Reference is seen being tracked with maximum error of 0.09 meters.

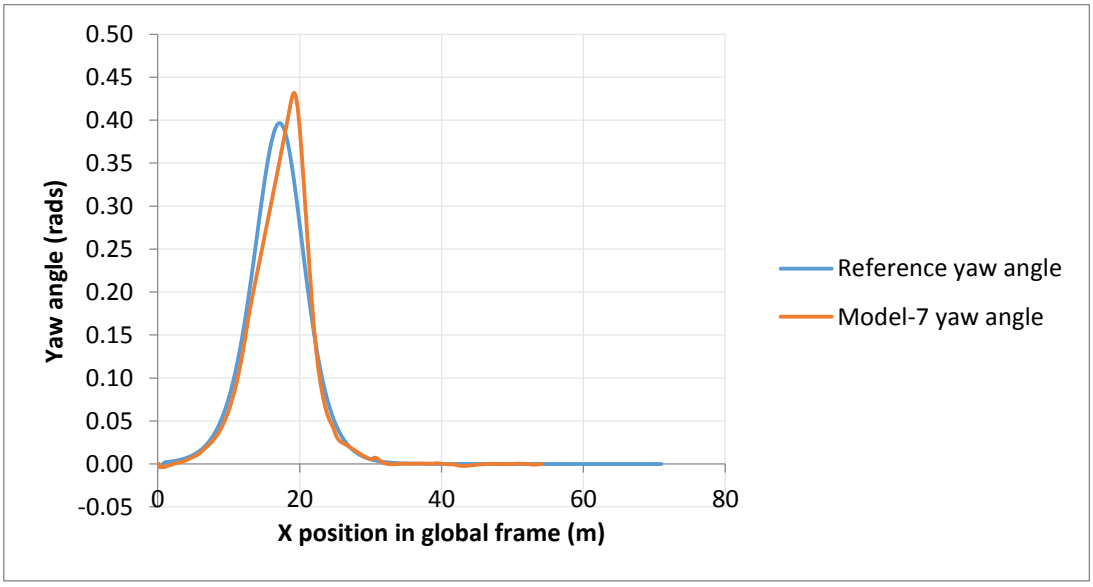

Figure 8.41: Showing reference yaw angle and model-7 yaw angle trajectories plotted against X position in global coordinate frame. Reference is seen being tracked with maximum error of 0.1 radians.

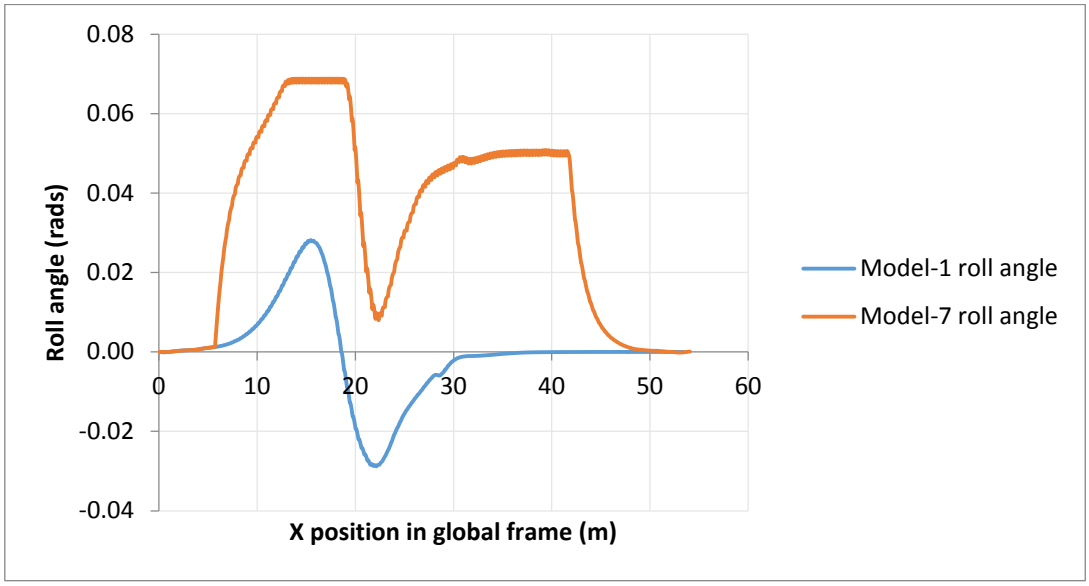

Figure 8.42: Showing Model-7 and Model-1 vehicle roll angles plotted against X position. Model-7 roll angle is seen shifted upwards between 5 and 42 meters of X position due to injected positive disturbance. Roll constraint (0.05 radians) is violated.

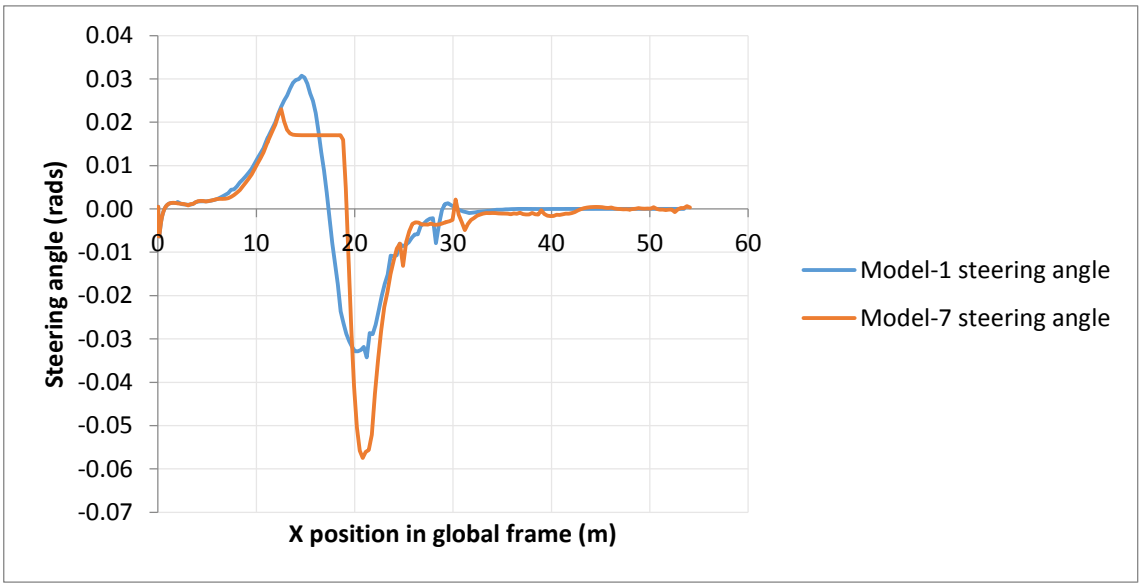

Figure 8.43: Showing Model-7 and Model-1 vehicle steering angles plotted against X position. The steering angles are seen to be different due the large (0.05 radians) disturbance in roll angle. Model-1 steering angle has peak values of 0.03 and -0.032 radians. Model-7 steering angle has peak values of 0.021 and -0.056 radians.

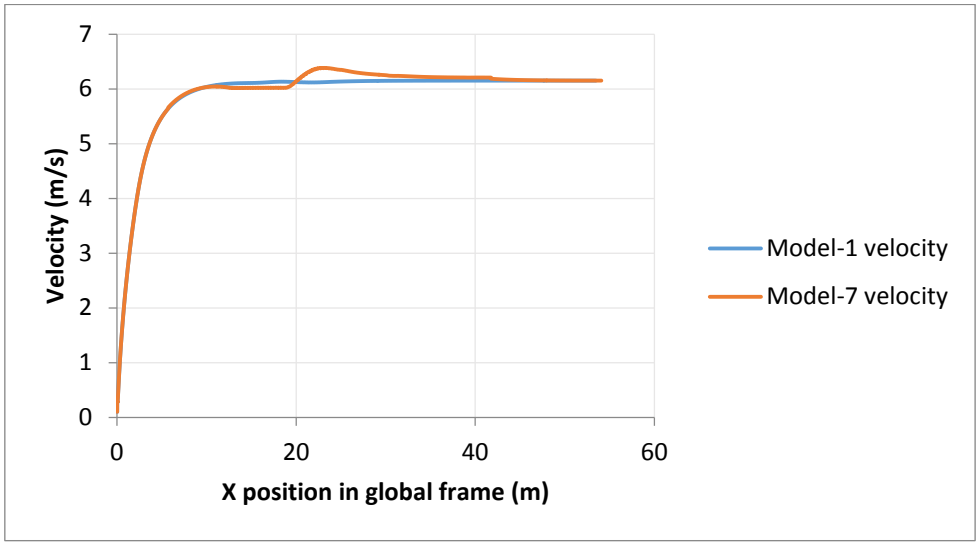

Figure 8.44: Showing Model-7 and Model-1 vehicle longitudinal velocities plotted against X position. Model-7 velocity tried to reduce between 10 and 20 meters of X position, this is same X position roll angle constraint was violated in figure 8.44. Velocity is seen rising to a peak value of 6.37 m/s after 20 meters of X position when roll angle was below the limit of 0.05 radians.

After introducing a 0.05 radians pulse disturbance in roll angle (figure 8.42), it is observed that the vehicle still tracks the reference YX and yaw angle trajectories as seen in figures 8.40 and 8.41 respectively. YX trajectory tracking maximum error is seen to be 0.09 meters, which is still less than maximum allowed value of 0.1 meters used in this study. The huge roll angle disturbance resulted in the roll angle of the vehicle being shifted upwards by almost 0.05 radians as can be seen in figure 8.42. In reference to model-1 (same figure 8.42), it is observed that the introduced roll disturbance pushed the vehicle's roll angle beyond its stability limit (roll angle constraint was violated).

From the disturbance simulation results, it is concluded that the designed controller is able to reject smaller roll angle disturbances (in this case 0.02 radians represented small disturbance in relation roll angle defined limit of 0.05 radians). During these small disturbances, the designed controller is able to track reference trajectory and prevent the vehicle from rollover event. This has been observed during simulation 5. In circumstances of huge disturbances (their maximum value was represented by 0.05 radians), the controller follows the reference trajectory but the roll angle constraint is violated as observed during simulation 6.

Violation of the roll angle constraint means the controller cannot prevent rollover event when faced with huge disturbances, in this case 0.05 radians represented huge disturbances. To deal with such disturbances (all those which the controller cannot reject) a nonlinear state estimator (e.g. Unscented Kalman filter) can be used to estimate

disturbance states as well as all other states. Nonlinear state estimators design is available in literature and is beyond the scope of this study.

8.4 Controller Tuning Parameters and Stability

The previous two sections investigated the designed controller's robustness and disturbance rejection capability. In this section tuning parameters of the controller, especially prediction horizon and number of control intervals are investigated. These two parameters are crucial in MPC control theory.

ACADO software solves MPC problems in least squares sense, the problem is presented as an optimal control problem (OCP). The OCP is called as follows:

$$OCP\ ocp(tStart, tEnd, N)$$

Where $tStart$ is the OCP start time, $tEnd$ is the OCP end time and N is the number of divisions (control intervals) between $tStart$ and $tEnd$. In this study $tStart$ was set to zero and $tEnd$ was the prediction horizon of the controller. In terms of number of divisions, prediction horizon was N and control horizon was $N - 1$. To investigate the effect of these parameters, four simulations were done as shown in table 8.3.

Table 8.3: Parameter values used during different simulations.

Simulation	**tStart (s)**	**tEnd (s)**	**N**	**Interval length (s) = (tEnd – tStart)/N**
Simulation 1	0.0	0.25	3	0.083
Simulation 2	0.0	0.50	5	0.1
Simulation 3	0.0	1.00	10	0.1
Simulation 4	0.0	2.00	30	0.067

Values for prediction horizon $(tEnd)$ were selected ranging from minimum possible value (0.25) to maximum possible value (2.00) with which the controller was feasible and able to follow the trajectory. The values for N were minimum possible number of divisions (or there about) with which the controller was feasible. Values of N greater than those in table 8.3 also work; producing similar results but at the expense of higher computation times. All other parameters of the controller were kept as previously described in this chapter.

The effect of these parameters was mainly seen on torque rate (thereby velocity) and solution times. Solution time is taken as time to complete one simulation loop (including simulation environment and external model as discussed in controller design chapter). Solution times can also be viewed as the difference in times at which control action is written to the plant. It was noted that most of this time is taken by solving of the OCP. Therefore, the results shown in the following graphs are torque rates, velocities and solution times. Torque rates and velocity graphs have been plotted against X position while the solution times graphs have been plotted against number of iteration (simulation loop).

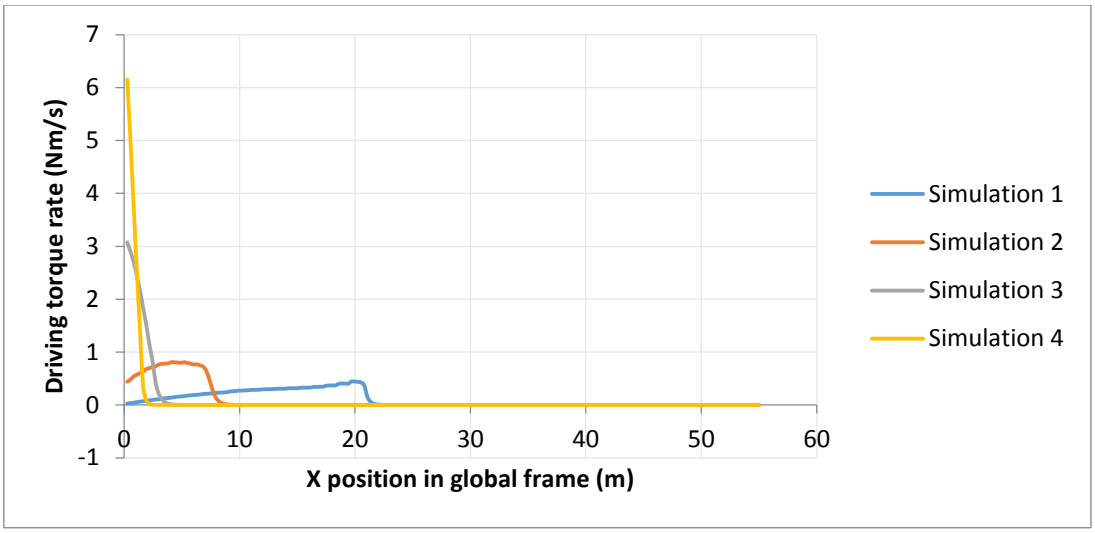

Figure 8.45: Showing driving torque rates for different simulation parameters. Aggressiveness is seen increasing from simulation 1 to simulation 4 accordingly.

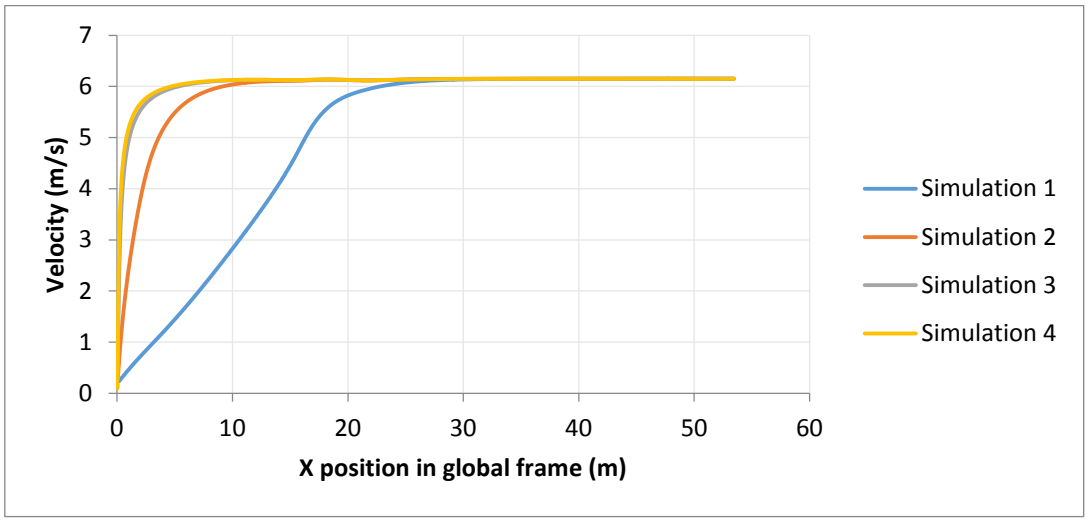

Figure 8.46: Showing velocities for different simulation parameters. Aggressiveness as seen in figure 8.45 results in aggressive velocities accordingly.

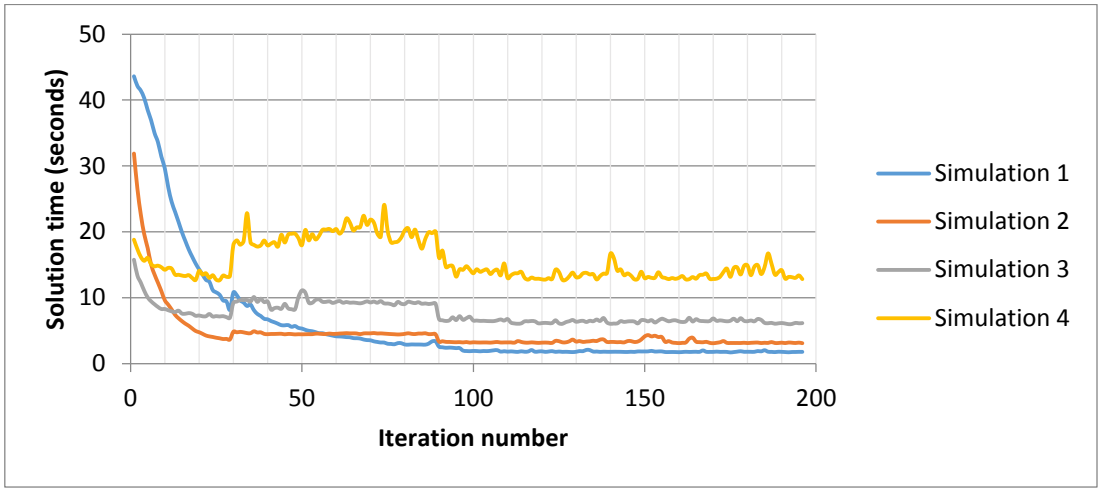

Figure 8.47: Showing solution times for different simulation parameters. Solution times are seen increasing from simulation 1 to simulation 4, except at start of simulation where simulation 1 has highest solution time followed by simulation 2, simulation 4, and simulation 3 in that order.

From the results in figures 8.45, 8.46 and 8.47, it is observed that as prediction horizon is increased (which increases number of control intervals subsequently), the controller achieves the maximum possible velocity quicker in reference to X position. The control actions are also seen to be moving from gentleness to aggressiveness as prediction horizon is increased. Similarly an increase in computation time is observed as number of control intervals (N) increases. It should be mentioned here that these results (velocities and control action) are heavily dependent on control divisions' number. The number of control divisions within the prediction horizon actually represents the number the OCP problem is solved within the horizon. Therefore, the higher the number (N), the closer the output to the set point (optimal value). These high values of N results in higher computation times as observed in the solution time graphs.

In other words, from the analysis of results it was observed that if number of control divisions (N) is kept constant, and only allow prediction horizon to vary, the controller became less aggressive as prediction horizon increased. This is the same explanation provided in the previous paragraph. Having shorter horizon, means within those N divisions (which are same for longer horizon); the controller achieves reference tracking quicker as compared to longer horizon. This causes the controller with shorter horizon to be aggressive if N is constant. As previously said, the choice of N depended on minimum value possible. The choices helped to determine the effect of N on computation times.

This chapter has presented results which show that the designed NMPC was able to track a given reference vehicle trajectory as well as respecting the state constraints. In terms of output, the constraints were mainly on roll angle and yaw-angle in this study. Roll angle constraint is directly related to vehicle rollover event as previously explained. If roll angle constraint is violated, it means rollover event has occurred. Input constraints were also imposed on driving torque and steering angle with their respective rates of change. None of the constraints were violated.

Trajectory tracking is seen to have improved greatly in comparison with the tracking errors that were observed in chapter three where related works are analysed. Tracking errors of 10 meters (on average) for full scale vehicles were observed in related works chapter. The controller designed in this study has limited the error to a maximum of 0.1 meters (on average) on scaled vehicle ($1/10^{th}$), which can be translated to limiting the error to maximum of 1 meter on full scale. This can be interpreted as a 10% improvement in terms of trajectory tracking capability.

In addition to tracking reference trajectories and respecting imposed constraints; robustness and disturbance rejection of the controller were also investigated. The controller showed robustness to parameter changes. It also showed its capability of rejecting smaller roll angle disturbances (0.02 radians) in relation to roll angle limit (0.05 radians). Lastly some tuning parameters were investigated in reference to mainly their effect on computational times. It was noted that increasing prediction horizon, in general reduces controller aggressiveness.

Chapter 9

Conclusions and Future Work

9.1 Conclusion

In this study, the goal was to develop an NMPC controller which is capable of maintaining autonomous vehicle stability by constraining yaw-rate, side-slip angle and roll angle (rollover prevention) while following a defined path at a maximum possible optimal speed. A nonlinear vehicle model which is capable of capturing all the dynamics of interest (yaw-rate, side-slip and roll angles) was used in the study. The parameters of the vehicle model were successfully identified using a scaled vehicle platform using grey box system identification methods: nonlinear least square in Scilab was used.

The NMPC controller was successfully designed using ACADO software, the controller was implemented within ACADO Toolkit environment not using code generation feature of ACADO. Closed loop simulations of the designed NMPC controller with the nonlinear vehicle model were done. The results showed that the controller is capable of following a defined trajectory without violating the constraints which were deduced through stability analysis of the vehicle model. In addition to trajectory following and rollover prevention by constraint respecting, robustness and disturbance rejection were also investigated. The results showed that the controller is robust and able to reject small disturbances throughout the path of travel.

From the study, it is concluded that the designed NMPC controller form is capable of driving the vehicle to follow a defined trajectory at the same time preventing yaw rate and rollover instability by observing and respecting the defined constraints. Furthermore, the controller can actually be implemented on actual physical system.

9.2 Future Work

This work could be extended to implementation on the actual vehicle platform using Linux-based micro-controllers like Jetson. After which direction towards making the vehicle fully autonomous could be taken, that is by incorporating more sensors. Furthermore, research could be done to investigate ways of reducing solution times (computational burden); either finding a means of taking advantage of ACADO code generation tool or parallel computing.

Bibliography

[1] N. H. T. S. Administration, “Traffic Safety Facts 2014: A Compilation of Motor Vehicle Crash Data from theFatality Analysis Reporting System and the General Estimates System,” NHTSA, Washington, 2014.

[2] L. Palkovics, A. Samsey and E. Gerum, “Roll-Over Prevention Systems for Commercial Vehicles - Additional Sensorless Function of Electronic Brake System,” *Vehicle System Dynamics,* vol. 32, no. 4, pp. 285-297, 1999.

[3] B. Chen and H. Peng, “Differential-Breaking Based Rollover Prevention for Sport Utility Vehicles with Human-in-loop Evaluations,” *Vehicle System Dynamics,* vol. 36, no. 4, pp. 359-389, 2001.

[4] T. Wielenga, “A method for reducing On-Road Rollovers: Ant-Rollover Braking,” in *SAE Technical Paper*, 1999.

[5] J. Ackermann and D. Odental, “Damping of vehicle roll dynamics by gain scheduled active steering,” in *1999 European Control Conference (ECC)*, Karlsruhe, Germany, 1999.

[6] S. Yim, “Design of a robust controller for rollover prevention with active suspension and differential braking,” *Journal of Mechanical Science and Technology,* vol. 26, no. 1, pp. 213-222, 2012.

[7] S. Solmaz, M. Carless and R. Shorten, “A Methodology for the Design of Robust Rollover Prevention Controllers for Automotive Vehicles: Part 1-Differential Braking,” in *Proceedings of the 45th IEEE Conference on Decision & Control*, CA, USA, 2006.

[8] T. Acarman and U. Ozguner, “Rollover Prevention for Heavy Trucks Using Frequency Shaped Sliding Mode Control,” *Vehicle System Dynamics,* vol. 44, no. 10, pp. 737-762, 2006.

[9] C. Parenteau, D. Viano, M. Shah, M. Gopal, J. Davies, D. Nichols and J. Broden, “Field Relevance of a suite of rollover test to real world crashes and injuries,” *Accident Analysis & Prevention,* vol. 35, no. 1, pp. 103-110, 2003.

[10] S. Solmaz, M. Corless and R. Shorten, “A methodology for the design of robust rollover prevention controllers for automotive vehicles with active steering,” *International Journal of Control,* vol. 80, no. 11, pp. 1763-1779, 2007.

[11] S. Jitesh, “ANTILOCK BRAKING SYSTEM (ABS),” *International Journal of Mechanical Engineering and Robotics Research,* vol. 3, no. 4, pp. 253-259, 2014.

[12] F. Borrelli, A. Bemporad, M. Fodor and D. Hrovat, “An MPC/Hybrid System Approach to Traction Control,” *IEEE Transactions on Control Systems Technology,* vol. 14, no. 3, pp. 541-552, 2006.

[13] E. Liebemann, K. Meder, J. Schuh and G. Nenninger, “Safety and Performance Enhancement: The Bosch Electronic Stability Control (ESP),” SAE International, Germany, 2004.

[14] N. H. T. S. Administration, “The Impact of Driver Inattention on Near-Crash/Crash Risk: An Analysis using 100-Car Naturalistic Driving Study Data,” National Technical Information Service, Virginia, 2006.

[15] E. Coelingh, A. Eidehall and M. Bengtsson, “Collision Warning with Full Auto Brake and Pedestrian Detection: A practical example of Automatic Emergency Braking,” in *13th International IEEE Conference on Intelligent Transportation Systems*, Funchal, 2010.

[16] R. Behringer, “The DARPA Grand Challenge - Development of an Autonomous Vehicle,” in *IEEE Intelligent Vehicles Symposium*, Parma, 2004.

[17] J. Levison, “Towards Fully Autonomous Driving: Systems and Algorithms,” in *2011 IEEE Intelligent Vehicles Symposium (IV)*, Baden-Baden, 2011.

[18] J. Yoon and K. Yi, “A Rollover Mitigation Control Scheme Based on Rollover Index,” in *Proceedings of the 2006 American Control Conference Minneapolis*, Minnesota, USA, June 14-16, 2006, pp. 5372-5377.

[19] J. Yoon, W. Cho, B. Koo and K. Yi, “Unified Chassis Control for Rollover Prevention and Lateral Stability,” *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY,* vol. 58, no. 2, pp. 596-609, 2009.

[20] Z. Tianjun and Z. Changfu, “Rollover Prevention for Heavy Trucks Using Robust Control,” in *2009 ISECS International Colloquium on Computing, Communication, Control, and Management*, Sanya, 2009.

[21] Z. Tianjun, W. Bingsheng and Z. Changfu, “The Design of Rollover Prevention System in Heavy Vehicles based on an Robust Method,” in *2009 2nd International Conference on Power Electronics and Intelligent Transportation System (PEITS)*, Shenzhen, 2009.

[22] S. Hong and J. Hedrick, “Roll Prediction-based Optimal Control for Safe Path Following,” in *2015 American Control Conference (ACC)*, Chicago, 2015, pp. 3261-3266..

[23] S. Qin and T. Badgwell, “An overview of industrial model predictive control technology,” *Chem Process Control,* vol. 93, no. 316, pp. 232-356, 1997.

[24] A. Bemporad, M. Morari, V. Dua and E. Pistikopoulos, “The explicit linear quadratic regulator for constrained systems,” *Automatica,* vol. 38, no. 1, pp. 3-20, 2002.

[25] M. Morari and J. Lee, “Model Predictive Control: past, present and future,” *Computers & Chemical Engineering,* vol. 23, no. 4-5, pp. 667-682, 1999.

[26] N. J. Noxon, “A model predictive control approach to roll stability of a scaled crash avoidance vehicle,” Masters dissertation, Califonia Polytechnic State University,

Califonia, 2012.

[27] M. Ghazali, M. Durali and H. Salarieh, “Path-following in model predictive rollover prevention using front steering and braking,” *Vehicle System Dynamics,* vol. 55, no. 1, pp. 121-148, 2017.

[28] H. Yuan, D. Zhang and T. Gordon, “Road vehicle rollover prevention torque vectoring via model predictive control,” in *Proceedings of the 36th Chinese Control Conference*, Dalian, 2017, pp. 9401-9406.

[29] S. Lee, F. Yakub, M. Kasahara and Y. Mori, “Rollover Prevention with Predictive Control of Differential Braking and Rear Wheel Steering,” in *2013 6th IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, Manila, 2013, pp. 144-149.

[30] K. Reif, Brakes, Brake Control and Driver Assistance Systems - Function, Regulation and Components, Springer, 2014.

[31] H. B. Pacejka, Tyre and Vehicle Dynamics, Elsevier LTD, 2006.

[32] B. E. Craig, “Applications of Model Predictive Control to Vehicle Dynamics for Active Safety and Stability,” PhD Dissertation, Stanford University, 2011.

[33] A. van Zanten, W. Ruf and A. Lutz, “Measurement and simulation of transient tire forces,” SAE Paper 890640, 1989.

[34] d. C. C. Wit, H. Olsson, K. J. Astrom and P. Lischinsky, “Dynamic Friction Models and Control Design,” in *1993 American Control Conference*, San Francisco, CA, USA, 1993, pp. 1920-1926.

[35] H. Dugoff, P. Fancher and L. Segel, “An Analysis of Tire Traction Properties and Their Influence on Vehicle Dynamic Performance,” SAE Technical Paper 700377, 1970.

[36] P. Helnwein, C. Liu, G. Meshke and H. Mang, “A new 3-D finite element model for cord-reinforced rubber composite:application to analysis of automobile tires,” *Finite Elements in Analysis and Design,* vol. 14, no. 1, pp. 1-16, 1993.

[37] M. Koishi, K. Kabe and M. Shiratori, “Tire Cornering Simulation Using an Expfieit Finite Element Analysis Code,” *Tire Science and Technology,* vol. 26, no. 2, pp. 109-119., 1998.

[38] A. van Zanten, W. Erhardt and A. Lutz, “ Measurement and simulation of transients in longitudinal and lateral tire forces,” SAE Paper 900210, 1990.

[39] T. Shim and C. Ghike, “Understanding the limitations of different vehicle models for roll dynamics studies,” *Vehicle System Dynamics,* vol. 45, no. 3, pp. 191-216, 2007.

[40] D. Murray-Smith, “Methods of System Identification, Parameter Estimation and Optimisation Applied to Problems of Modelling and Control in Engineering and Physiology,” DSc thesis, University of Glasgow, Glasgow, 2009.

[41] L. Ljung and G. T., Modelling of dynamic systems, Englewood Cliffs: Prentice Hall, 1994.

[42] O. Nelles, Nonlinear System Identification, Berlin, Germany: Springer, 2001.

[43] S. Kirkpatrick, C. Gelatt and M. Vecchi, “Optimization by simulated annealing,” *Science,* vol. 220, no. 4598, pp. 671-680, 1983.

[44] P. Van Laarhoven and E. Aarts, Simulated Annealing: Theory and Applications, Dordrecht, Lancaster: Springer, 1987.

[45] D. Goldberg, Genetic Algorithms in Searching, Optimisation and Machine Learning, Reading, USA: Addison-Wesley, 1989.

[46] T. Söderström and P. Stoica, System Identification, New York: Prentice-Hall, 1989.

[47] B. Schofield, “Vehicle Dynamics Control for Rollover Prevention,” PhD thesis, Department of Automatic Control, Lund Institute of Technology, Lund University, 2006.

[48] S. Rakheja and A. Piche, “Development of Directional Stability Criteria for an Early Warning Safety Device,” *JOURNAL OF COMMERCIAL VEHICLES,* vol. 99, no. 2, pp. 877-889, 1990.

[49] P. Thomas and J. H. F. Woodrooffe, A feasibility study of a rollover warning device for heavy trucks., Canada: WordCat, 1990.

[50] E. Dahlberg, “Commercial Vehicle Satbility - Focusing on Rollover,” PhD thesis, Royal Institute of Technology, Stockholm, 2001.

[51] B. Johansson and M. Gafvert, “Untripped SUV Rollover Detection and Prevention,” in *43rd IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, 2004.

[52] P. Liu, S. Rakheja and A. Ahmed, “Dynamic rollover threshold of articulated freight vehicles,” *International Journal of Heavy Vehicle Systems,* vol. 5, no. 4, pp. 300-322, 1998.

[53] J. Rawlings and D. Mayne, Model Predictive Control: Theory and Design, Nob Hill, 2009.

[54] B. Houska, J. Ferreau H and M. Diehl, “ACADO Toolkit - An Open Source Framework for Automatic Control and Dynamic Optimisation,” *Optimal Control Applications and Methods,* vol. 32, no. 3, pp. 298-312, 2011.

[55] A. Kozma, J. Andersson, C. Savorgnan and M. Diehl, “Distributed Multiple Shooting for Optimal Control of Large Interconnected Systems,” *IFAC Proceedings Volumes,* vol. 45, no. 15, pp. 143-147, 2012.

[56] M. Diehl, H. Ferreau and N. Harvebeke, “Nonlinear model predictive control, volume 384 of Lecture Notes in control and Information Science,” Springer, 2009.

[57] L. Grune and J. Pannek, Nonlinear Model Predictive Control, Springer, 2011.

[58] S. Keerthi and E. Gilbert, “Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: stability and moving horizon approximations,” *Journal of Optimization Theory and Applications,* vol. 57, no. 2, pp. 265-293, 1988.

[59] H. Chen and A. F., “A Quasi-InÞnite Horizon Nonlinear Model Predictive Control Scheme with Guaranteed Stability,” *Automatica,* vol. 34, no. 10, pp. 1205-1217, 1998.

[60] A. Watchter, “An Interior Point Algorithm for Large-Scale Nonlinear Optimisation with Applications in Process Engineering,” PhD thesis, Carnegie Mellon University, Pittsburgh, 2002.

[61] T. Optimisation, “Matlab Optimal Control Software(ODE,DAE),” http://tomdyn.com, 2009-2011.

[62] B. Fabien, “dsoa: The implementation of a dynamic system optimisation algorithn,” *Optimal Control Applications and Methods,* vol. 31, no. 3, p. 231–247, 2009.

[63] D. Leineweber, I. Bauer, H. Bock and J. Schloder, “An Efficient Multiple Shooting Based Reduced SQP Strategy for Large-Scale Dynamic Process Optimisation. Part I: Theoretical Aspects,” *Computers & Chemical Engineering,* vol. 27, no. 15, pp. 157-166, 2003.

[64] L. Simon, Z. Nagy and K. Hungerbuehler, “Nonlinear Model Predictive Control, Volume 384 of Lecture Notes in Control and Information Sciences: OptCon,” Springer, 2009.

[65] R. Findeisen, F. Allogower and L. Biegler, Assessment and Future Directions of Nonlinear Model Predictive Control, Springer, 2006, pp. 229-238.

[66] S. L. Campbell, P.-J. Chancelier and R. Nikoukhah, Modelling and Simulation in Scilab/Scios, Springer, 2000.

[67] A. Sehati, “Modeling and Identification of Urban Bus Dynamics,” Masters report, Eindhoven University of Techmnology, 2016.

[68] E. a. C. Support, Body Builder Information Book, 2007.

[69] R. Quirynen, S. Gros and M. Dehl, “Fast auto generated ACADO integrators and application to MHE with multi-rate measurements,” in *2013 European Control Conference (ECC)*, Zurich, 2013, pp. 3077-3082.

[70] M. Vukov, W. V. Loock, B. Houska, H. Ferreau, J. Swevers and M. Diehl, “Experimental Validation of Nonlinear MPC on an Overhead Crane using Automatic Code Generation,” in *2012 American Control Conference (ACC)*, Montréal, 2012, pp. 6264-6269.

[71] F. Borrell and M. Morari, “Offset Free Model Predictive Control,” LA, USA, 2007.

[72] M. Wang, Y. Kung, C. Chiang and Y. Wang, “Permanent magnet linear synchronous motor drive design based on sliding mode control and fuzzy deadzone estimation,” in *2009 IEEE International Conference on Systems, Man and Cybernetics*, Antonio, 2009, pp. 1027-1032.

[73] J. Jang, “A deadzone compensator of a dc motor system using fuzzy logic control,” in *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 2001, pp. 42-48.

[74] T. Senjyu, T. Yoshida, K. Uezato and T. Funabashi, “Position control of ultrasonic motors using adaptive backstepping control and deadzone compensation with fuzzy inference,” in *2002 IEEE International Conference on Industrial Technology*, Bankok, 2002, pp. 560-565 .

[75] K. Hsu, W. Wang and P. Lin, “Sliding mode control for uncertain nonlinear systems with multiple inputs containing sector nonlinearities and deadzones,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics),* vol. 34, no. 1, pp. 374-380, 2004.

[76] X. Sun, W. Zhang and Y. Jin, “Stable adaptive control of backlash nonlinear systems with bounded disturbances,” in *Proc.of the 31st Conference on Decision and Control*, Tucson, AZ, USA, 1992 pp. 274-275.

[77] N. Ahmad, H. Ebraheem, M. Alnaser and J. Alostath, “Adaptive Control of a DC Motor with Uncertain Deadzone Nonlinear at the Input,” in *2011 Chinese Control and Decision Conference (CCDC)*, Mianyang, 2011, pp. 4295-4299..

[78] P. Falcone, F. Borrelli, H. E. Tseng, J. Asgari and D. Hrovat, “Linear Time Varying Model Predictive Control and its Application to Active Steering Systems: Stability Analysis and Experimental Validation,” *Int. J. Robust Nonlinear Control,* vol. 18, no. 1, pp. 862-875, 2007.

[79] J. W. R. Howe and G. Forkenbrock, “An Experimental Examination of Selected Maneuvers That May Induce On-Road,” National Highway Traffic Safety Administration, 2002.

Appendix A

A. Additional Materials

A.1 Robustness Test Graphs

The graphs showing yaw rate, roll rate, steering rate and driving torque rate for the NMPC controller robustness test.

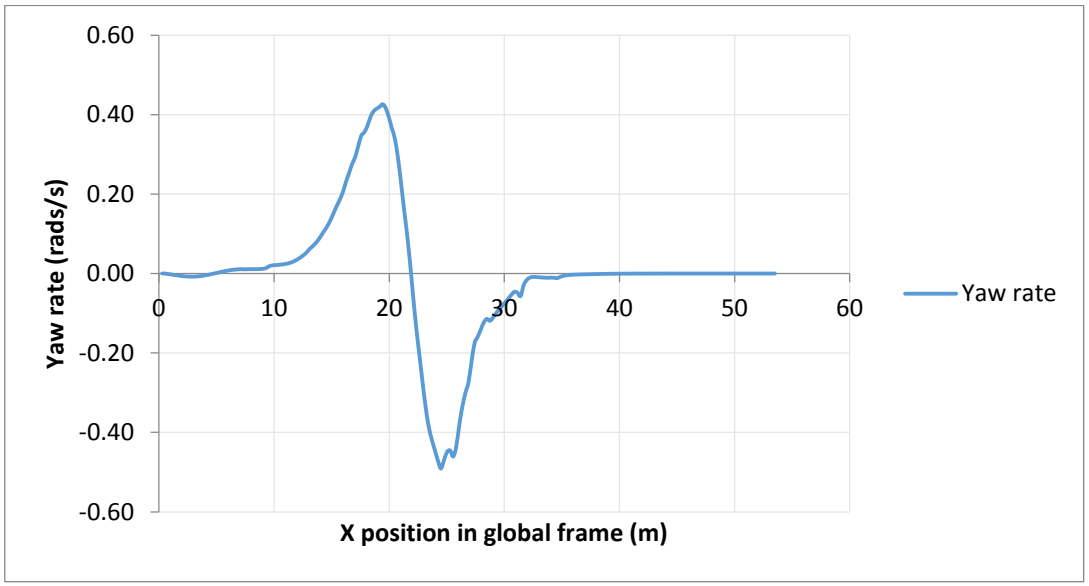

Figure A.1: Showing Model-2 yaw rate plotted against X position.

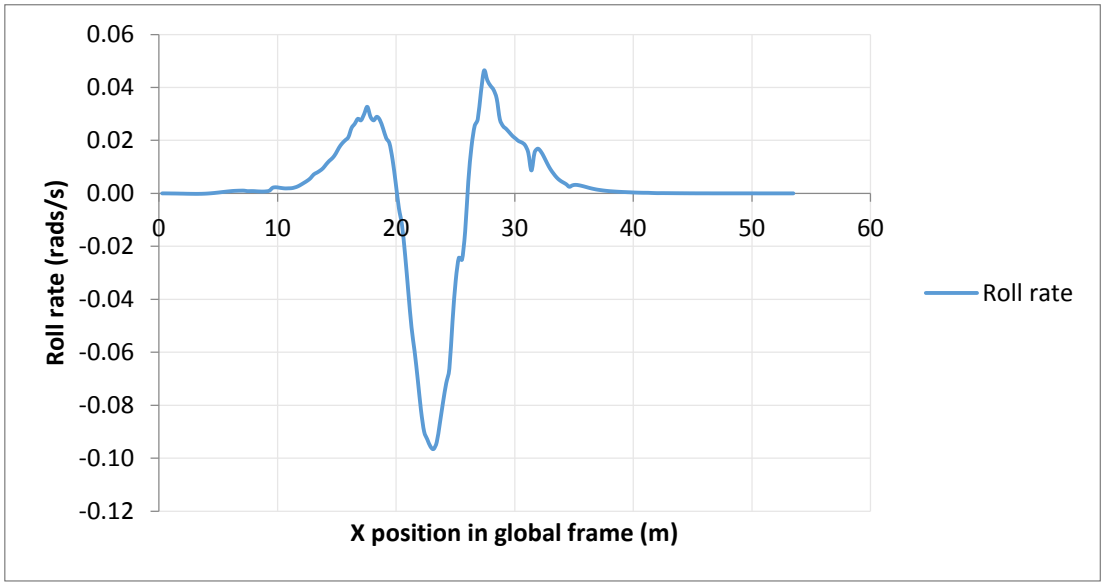

Figure A.2: Showing Model-2 roll rate plotted against X position.

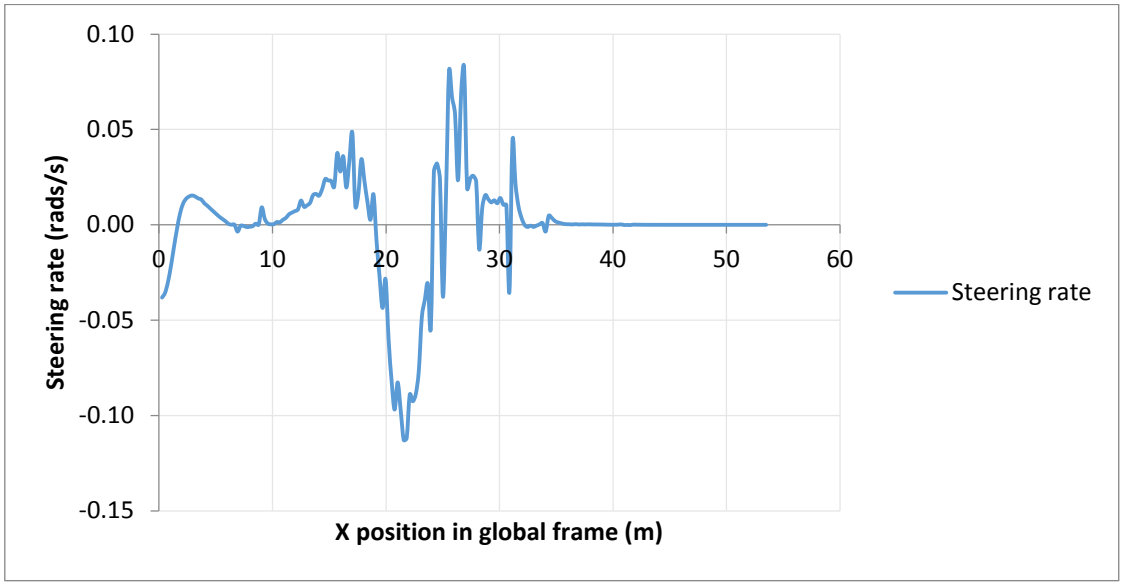

Figure A.3: Showing Model-2 steering rate plotted against X position.

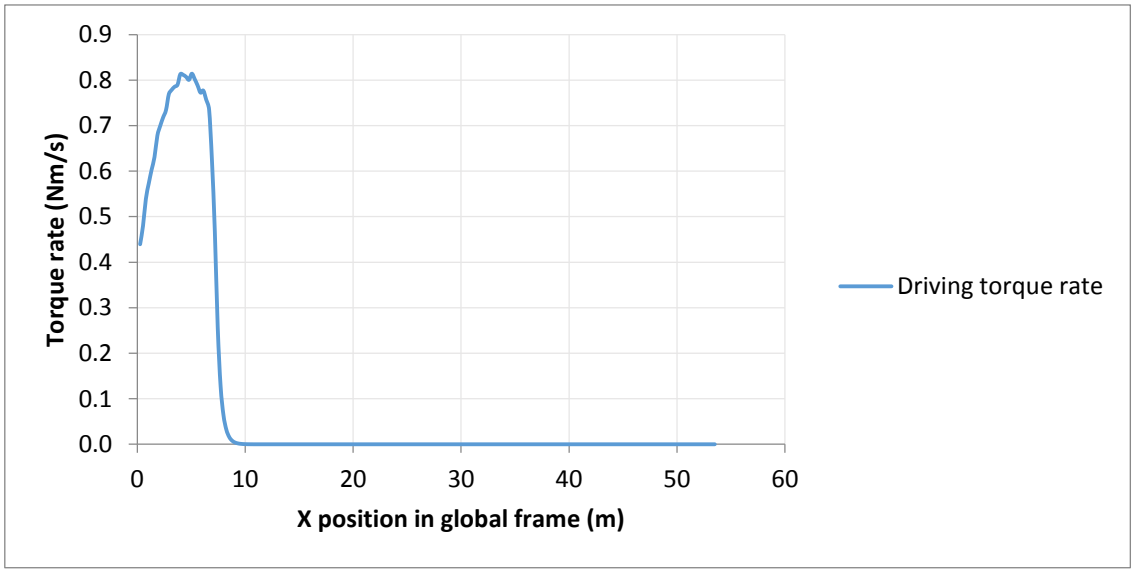

Figure A.4: Showing Model-2 driving torque rate plotted against X position.

A.2 Disturbance Rejection Test Graphs

The graphs showing yaw-rate, steering rate and driving torque rate for the NMPC controller disturbance rejection test.

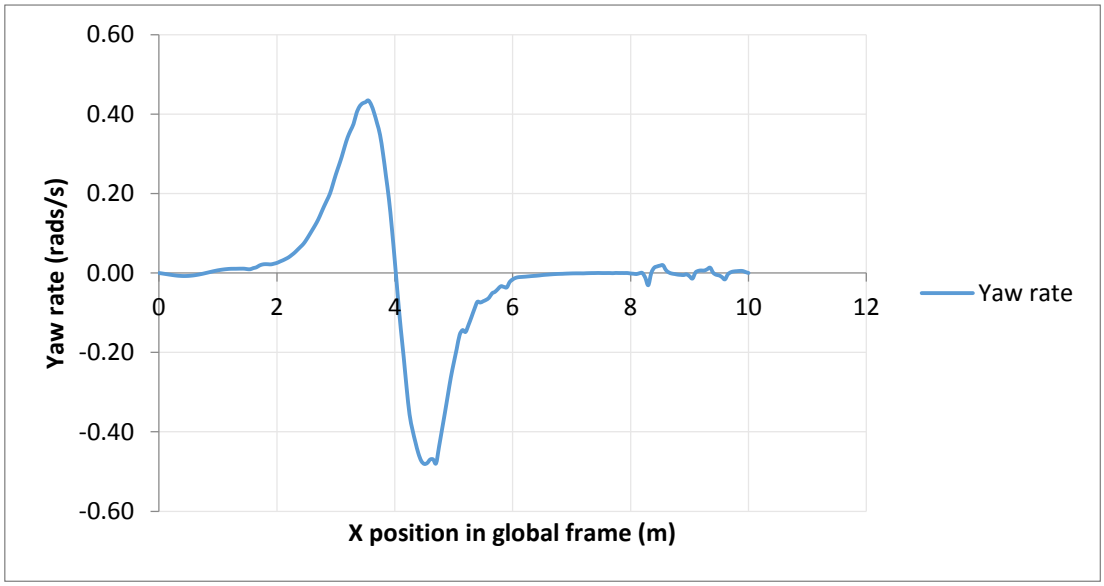

Figure A.5: Showing Model-6 yaw rate plotted against X position.

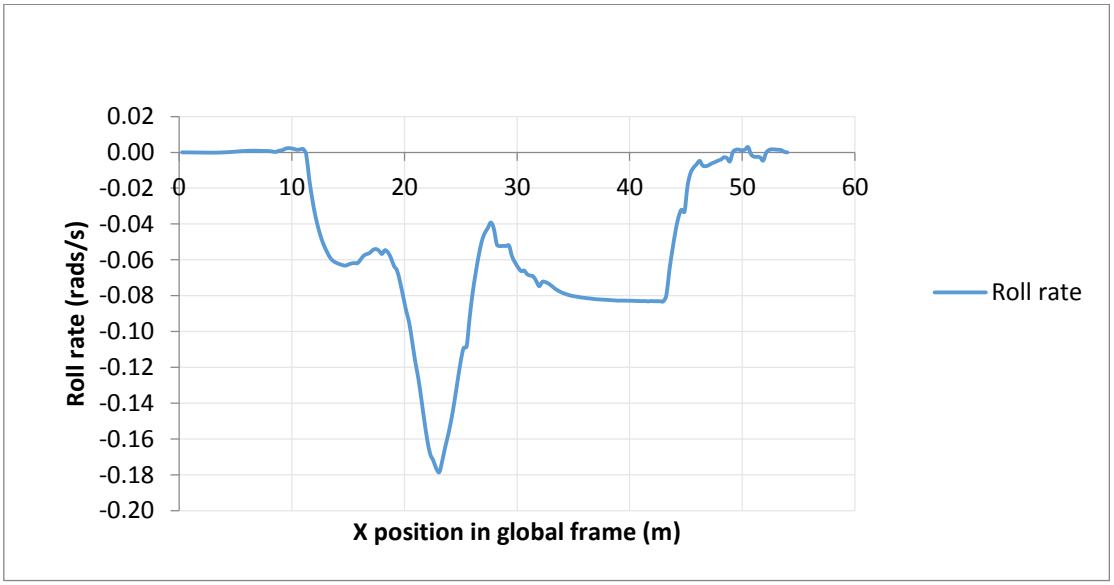

Figure A.6: Showing Model-6 roll rate plotted against X position.

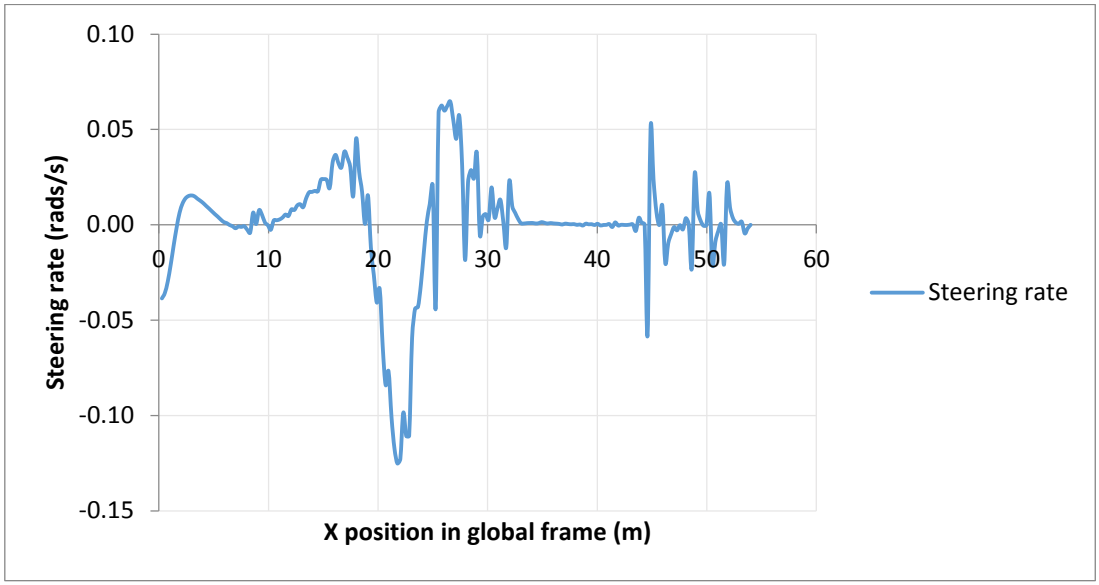

Figure A.7: Showing Model-6 steering rate plotted against X position.

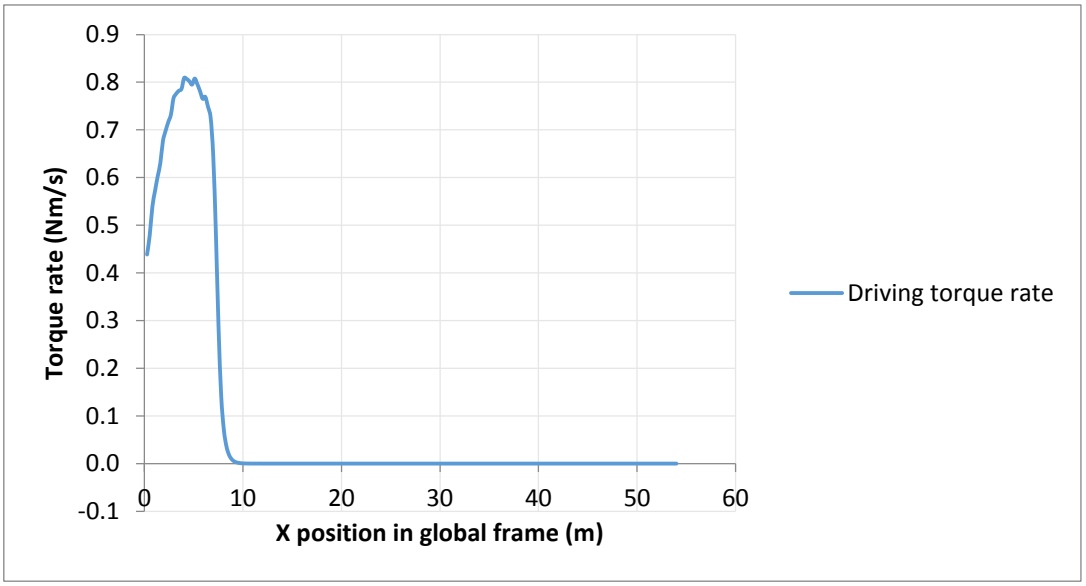

Figure A.8: Showing Model-6 driving torque rate plotted against X position.

Appendix B

B. Additional Materials

B.1 Platform Vehicle Model Parameters

Table B.1: Least Squares generated vehicle parameters

PARAMETER	VALUE
Longitudinal distance between the center of tire and vehicle center of mass	$l_f = 0.1810181\ m$ $l_r = 0.7400993\ m$
Front and rear track width	$t_f = 0.16282499\ m$ $t_r = 0.16282499\ m$
Distance between ground and static roll center	$h_f = 0.1742567\ m$ $h_r = 0.0760477\ m$
Height of sprung mass CG above ground	$h_{cgs} = 0.1766831\ m$
Total/Sprung vehicle mass	$M = 3.1007623\ kg$ $M_s = 2.0056593\ kg$
Front/Rear unsprung mass	$M_{uf} = 0.7924513\ kg$ $M_{ur} = 0.3026517\ kg$
Moments of inertia of sprung mass about its CG	$I_{xxs} = 0.0056456\ kg.m^2$ $I_{zzs} = 0.0506463\ kg.m^2$
Moment of inertia of wheel about its spin axis	$I_{tlf} = 0.0018781\ kg.m^2$ $I_{trf} = 0.0018781\ kg.m^2$ $I_{tlr} = 0.0018781\ kg.m^2$ $I_{trr} = 0.0018781\ kg.m^2$
Front/rear suspension spring stiffness	$K_{sf} = 15.999938\ N/m$ $K_{sr} = 10.999901\ N/m$
Front/rear shock damping	$B_{sf} = 2.999838\ N.s/m$ $B_{sr} = 1.999896\ N.s/m$
Front/rear wheel damping	$B_{wf} = 0.0099039\ N.s/m$ $B_{wr} = 0.0099039\ N.s/m$
Tire Radius	$r_w = 0.0612775\ m$
Cornering Stiffness and Longitudinal Stiffness	$C_{c\alpha} = 26.07626\ N\ deg^{-1}$ $C_{ck} = 185.2848\ N\ deg^{-1}$

Appendix C

C. NMPC Controller for DC Motor

The NMPC controller was designed in chapter 6 of this report. Before taking the proposed NMPC controller form to the derived vehicle model, which is a very high order system, it was decided that the controller be first tested on a simpler system. A simpler system would provide much understanding of the controller performance by making controller parameter adjustment a bit simple due to fewer states number. A DC motor system was chosen as a candidate for testing the proposed NMPC controller form designed in chapter 6.

Therefore, in this appendix the focus is on the implementation of the proposed NMPC controller on a DC motor to control velocity and position. The simulation and implementation results have been presented in this appendix. The appendix begins by modelling the DC motor and proceeds to simulating and implementing the controller.

C.1 DC Motor Modelling and Identification

This section is dedicated to modelling and identification of the DC motor which has been used as a case study in this study. The DC motor used in this research is available in Control and instrumentation laboratory at the University of Cape Town.

The open loop system, which is named the DCM150F DC motor and its associated components, is shown in figure C.1.

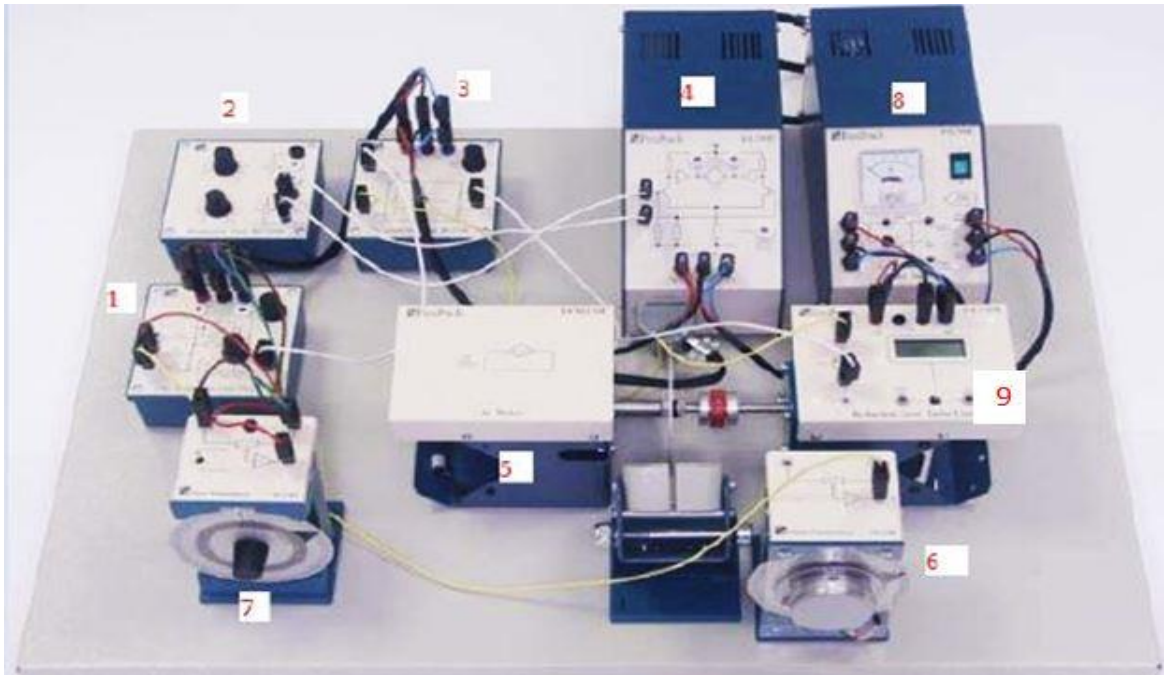

Figure C.1: DC Motor and its associated components.

The components shown in the open loop system are listed as follows:

1- The Operational Amplifier Unit OA150A.
2- The attenuator Unit AU150B.
3- The Pre-amplifier Unit PA150C.
4- The Servo Amplifier Unit SA150D.
5- The DC Motor Unit DCM150F.
6- The Output Potentiometer Unit OP150K.
7- The Input Potentiometer Unit IP150H.
8- The Power Supply PS150E.
9- The Tachometer Unit GT150X.

The block diagram of the open loop DC Motor system can be drawn as shown in figure C.2:

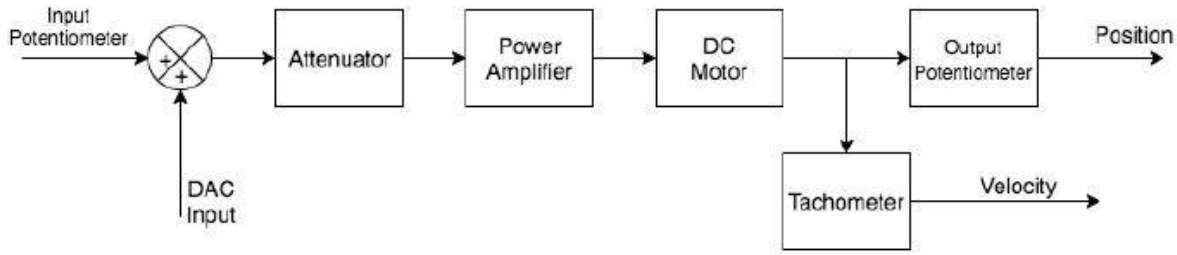

Figure C.2: DC Motor System open loop block diagram.

Input to the system is voltage which is supplied by adjusting input potentiometer; output can either be velocity or position. Velocity is obtained by measuring voltage on the tachometer output while position is obtained by measuring voltage across the output potentiometer. In calculations, input is voltage in volts (V) and output is either velocity in radian per second (rads/s) or position in radians (rads). In case of feedback connection, the feedback signal is connected through the digital analog converter (DAC) as shown in the block diagram. The main component of the system is the DC motor itself, hence more emphasis in the next paragraph is on modeling of this DC motor unit. The other components are lumped together to give a single gain value as explained later in this section.

Motor Modelling

The DC Motor Unit DCM150F is the main component in the open loop system. It is an armature controlled DC motor and its schematic representation can be seen in figure C.3.

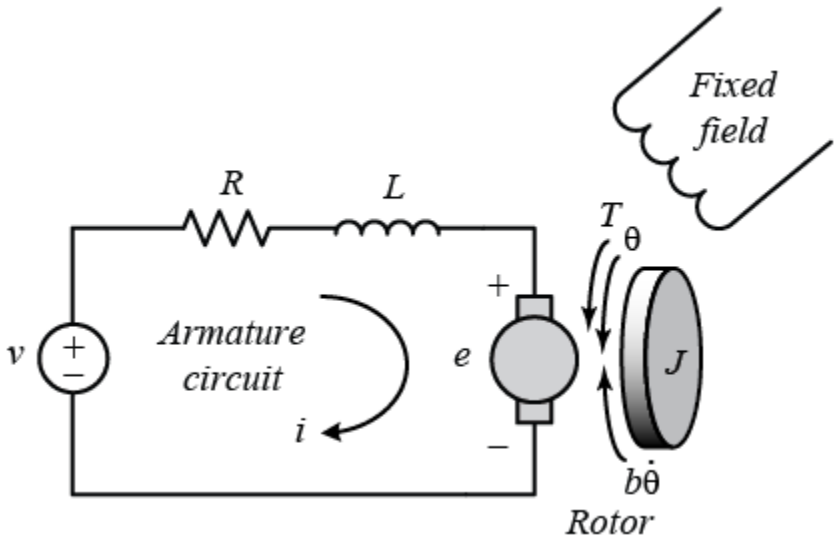

Figure C.3: Armature controlled DC motor diagram.

In figure C.3, v and e are the applied voltage and back emf respectively of the motor. R, L, T, J and b are resistance, inductance, torque, rotational inertia and damping coefficient respectively of the motor. Position and speed of the motor are represented by θ and $\dot{\theta}$ respectively. Using the diagram and assuming that the electrical time constant $\frac{L}{R}$ is very small compared to mechanical time constant $\frac{J}{b}$, the transfer function of the DC motor can be obtained as follows:

The mechanical torque T and the back emf e are given by:

$$e = K_e\dot{\theta} = v - L\frac{di}{dt} - Ri, \tag{C.1}$$

$$T = K_t i = J\ddot{\theta} + b\dot{\theta}. \tag{C.2}$$

Where K_e and K_t are back emf constant and motor constant respectively. The current flowing in the circuit is i. Rearranging the equations and taking Laplace transform leads to the following equations:

$$(Ls + R)I(s) = V(s) - K_e\dot{\theta}(s), \tag{C.3}$$

$$K_t I(s) = (Js + B)\dot{\theta}(s). \tag{C.4}$$

Combining the two equations (C.3 and C.4) by eliminating $I(s)$ from them, and dividing by $V(s)$, the speed transfer function of the DC motor is obtained as:

$$P_v(s) = \frac{\dot{\theta}(s)}{V(s)} = \frac{K_t}{(Ls+R)(Js+B)+K_tK_e} \qquad \left[\frac{rad/sec}{V}\right]. \tag{C.5}$$

Based on the earlier assumptions, the speed transfer function can be simplified to:

$$P_v(s) = \frac{K_m}{\tau s+1} \qquad \left[\frac{rad/sec}{V}\right]. \tag{C.6}$$

Where $\tau = \frac{J}{B}$ and K_m is overall motor constant taking into account K_t and K_e. The position transfer function is obtained by integrating equation C.6 with respect to time. The position transfer function is written as:

$$P_p(s) = \frac{K_m}{s(\tau s+1)} \qquad \left[\frac{rad}{V}\right]. \tag{C.7}$$

Motor System Identification

The next step was to identify the parameters K_m and τ of transfer function of the DC motor system presented. To identify these parameters, the speed (velocity) transfer function (equation C.6) was used. It was used because it is Bounded Input Bounded Output (BIBO) stable. To find the parameters, step inputs of different magnitude were applied to the physical motor system and output velocity (represented by tachometer output voltage) logged into a file for analysis. But before applying the steps, it was necessary to consider nonlinearities in DC motors, which would then influence the decision on magnitude of step inputs for proper and correct data collection.

The most two common nonlinearities in DC motors are deadzone and output saturation. Deadzone refers to the region around 0 *V* in the DC motor's transfer function where the input voltage applied is not enough to overcome inertia and the motor's own internal friction. Saturation refers to case in which an increase in the input voltage V does not have a resultant effect on the angular velocity $\dot{\theta}$. To determine deadzone and saturation of the DC motor used in this study, two ramp inputs (positive and negative) were supplied to the motor and the output velocities were recorded. The idea of applying two ramps was thought of as a way of taking into account that the deadzone may not be symmetrical, as well as considering the fact that the deadzone as seen from one direction may be different as seen from the opposite direction due to motor motion before it comes to rest. The two ramp inputs are shown in figure C.4.

The two ramp inputs named input voltage 1 and input voltage 2 in the figure C.4 are given by $v1 = 2.68 - 0.2 * t$ and $v2 = -2.68 + 0.2 * t$ respectively, where t is time in seconds.

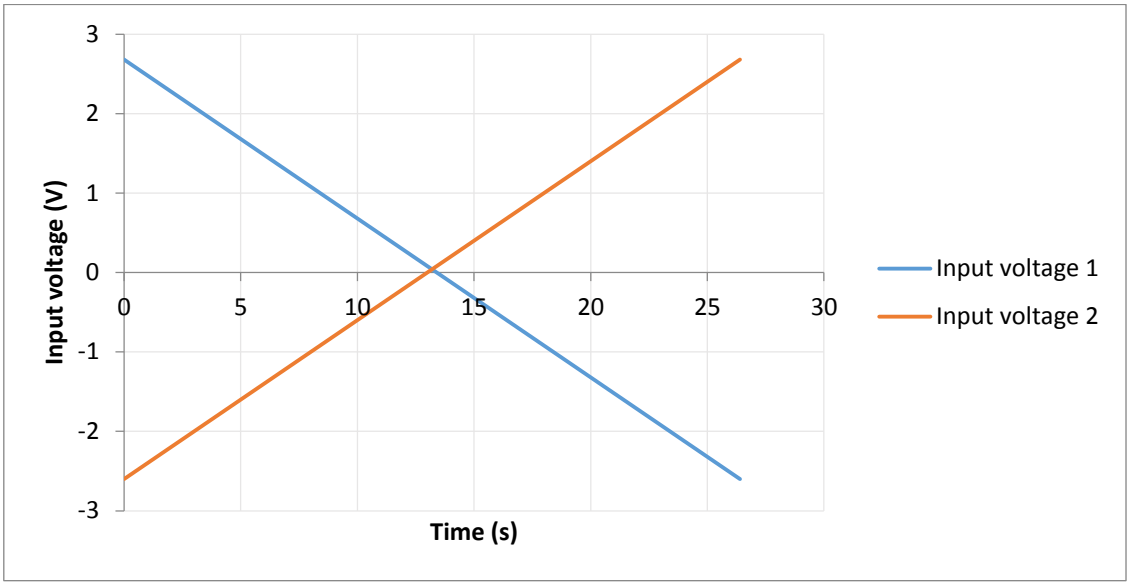

Figure C.4: Showing two ramp input voltages applied to the DC motor to identify motor nonlinearities.

Figure C.5 shows the outputs of the motor after applying the two ramps shown in figure C.4. The two outputs have been plotted on the same voltage axis for easy interpretation.

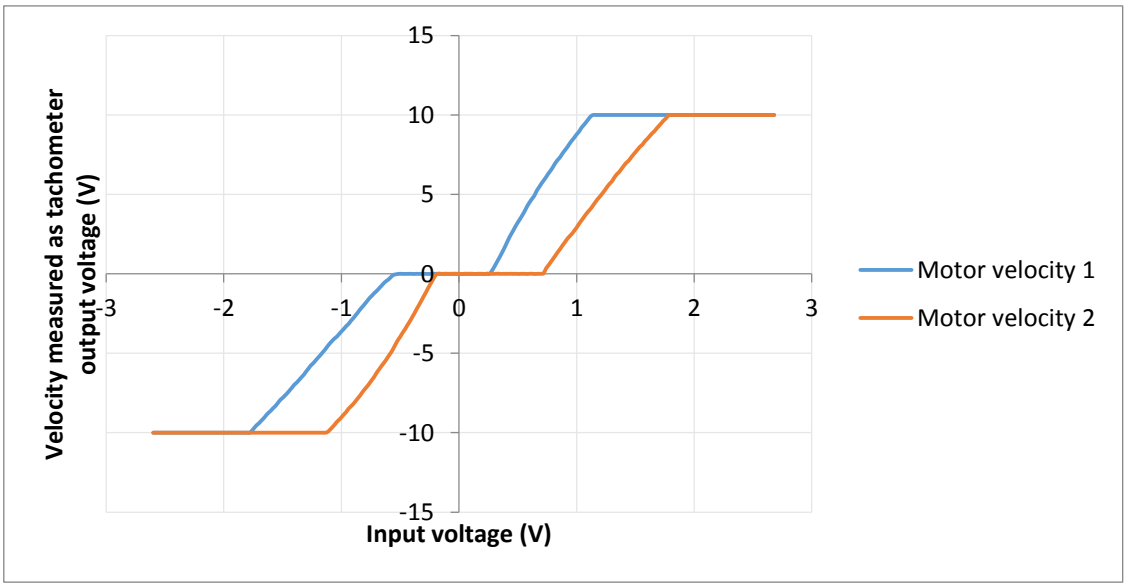

Figure C.5: Showing motor velocities as functions of input ramp voltages v1 and v2 shown in figure C.4.

Ramp input voltage 1 results in motor velocity 1 and ramp input voltage 2 results in motor velocity 2 on figure C.5. From the figure, it is seen that the motor's saturation limits are at 1.76 and -1.76 Volts. The motor's deadzone is between -0.5 and 0.64 Volts. These results are summarised in the table as follows:

Table C.1: Nonlinear characteristics of the DC motor.

Nonlinearity	**Region of effect**
Deadzone	-0.5 V to 0.64 V
Saturation	< -1.76 V and >1.76 V

After determining the motor's nonlinear regions, step inputs (positive and negative) were applied to the motor in order to determine the transfer characteristics between input and output. The final values of the parameters of the velocity transfer function were taken as average of the many tests that were performed. For the sake of brevity and clarity, two step responses (positive and negative) have been plotted here, which illustrate how the tests were performed. Figure C.6 shows a positive step response, and figure C.7 shows a negative step response of the DC motor system. The output velocity and step input voltage have been plotted on same graphs to ease readability.

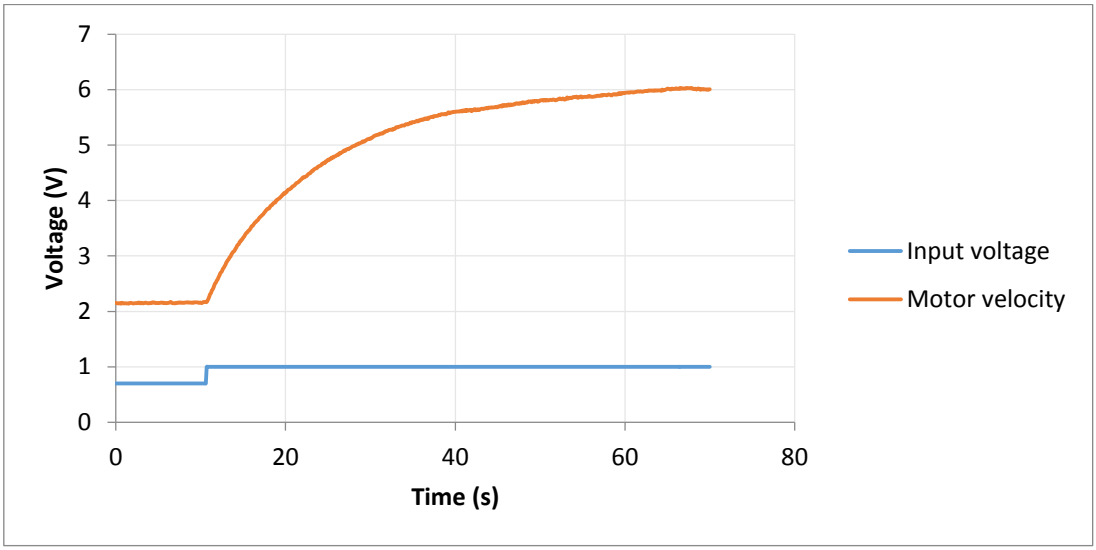

Figure C.6: Showing positive step (0.3 V) response of the DC motor system.

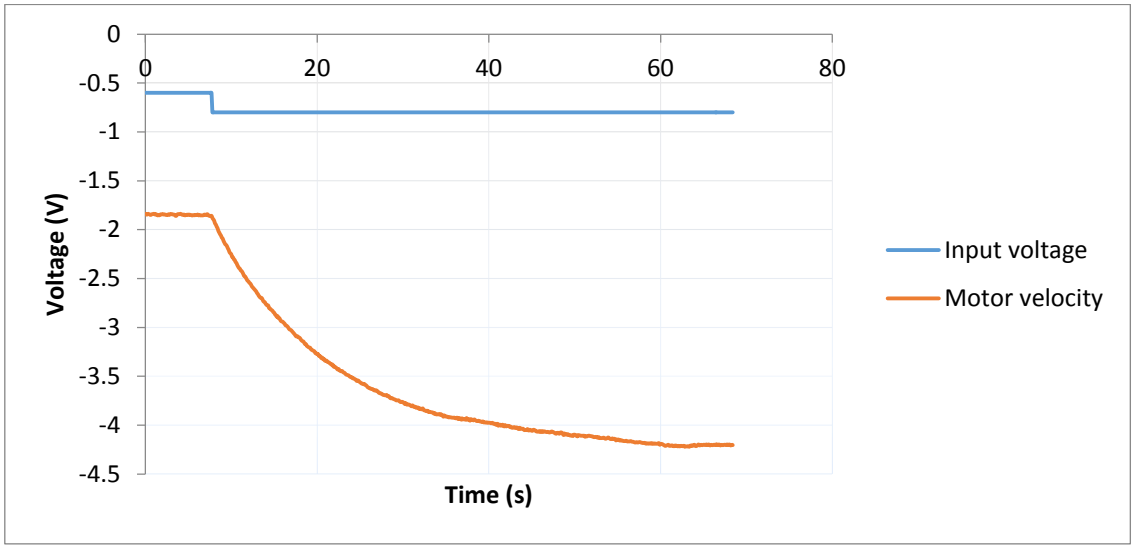

Figure C.7: Showing negative step (-0.2 V) response of the DC motor system.

As seen in figures C.6 and C.7, the motor was allowed to rotate at constant velocity by applying fixed inputs of 0.7 V and -0.6 V respectively. The aim was to reduce the effect of deadzone by keeping the motor in motion first before applying a step. A step of 0.3 V (positive step) was applied to the motor and the response can be seen on the same figure C.6. Similarly, a step of -0.2 V (negative step) was applied to the motor and the response can be seen on the same figure C.7.

Modeling the input step as a function $u(s) = \frac{B}{s}$ where B is the amplitude of the step allows the final value $y(\infty)$ to be calculated as $y(\infty) - y(0) = k_m B$, given $\lim_{t\to\infty} y(t) = \lim_{s\to 0} sY(s)$ where $Y(s) = u(s)P_v(s)$. For the response in figure 7.6 (positive step); B = 0.3 and $y(\infty) - y(0) = 6 - 2.15 = 3.85$, which gives $k_m = \frac{3.85}{0.3} = 12.8$. For the value of τ, which is defined as the 63% rise time in relation to the steady state value, in the figure 7.6 (positive step), it refers to the time when output reaches 63% of 3.85 plus 2.15. The time constant can be read off from the figure as 12.9 seconds approximately. Similar calculations can be performed for the negative step response. From a number of step responses that were performed, the average value for k_m was found to be 12 and τ was 13 seconds. This resulted into a nominal transfer function for the DC motor position to be

$$P_p(s) = \frac{12}{s(13s+1)} \quad \left[\frac{rad}{V}\right]. \tag{C.8}$$

C.2 DC Motor Simulation Results

The controller form designed in chapter 6 of this report was simulated in closed loop form with the DC motor model derived in this appendix. This section presents simulation results of velocity control as well as position control. For continuity purposes, the controller equation is presented here again as follows:

$$\underset{\xi(.),\ u(.)}{\text{minimise}} \quad \int_0^T \left(\left\| y(t) - y_{ref}(t) \right\|_Q^2 + \left\| u(t) - u_{ref}(t) \right\|_R^2 \right) dt \tag{C.9a}$$

$$\text{subject to} \quad \xi(0) = \xi_0 \,, \tag{C.9b}$$

$$\forall t \in [0,T]: \quad \dot{\xi}(t) = f\left(t, \xi(t), u(t)\right), \tag{C.9c}$$

$$\forall t \in [0,T]: \quad y(t) = h\left(t, \xi(t), u(t)\right), \tag{C.9d}$$

$$\forall t \in [0,T]: \quad \underline{u} \leq u(t) \leq \bar{u}, \tag{C.9e}$$

$$\forall t \in [0,T]: \quad \underline{\xi} \leq \xi(t) \leq \overline{\xi}. \tag{C.9f}$$

All the variables in this equation C.9 have already been explained in chapter 6. The choice of prediction horizon T and number of control intervals, defined by $\frac{T}{t}$, was analysed in detail in chapter 8. In this appendix, the choice was determined by the implementation results, which influenced the simulation parameters choice so that they at least match for ease of comparison. In this study, standard MPC formulation has been used where control horizon length is one less than prediction horizon length.

To fit the DC motor equation into this controller form, position transfer function of equation C.8 was first written in state space form as follows:

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= (-x_2 + 12(v))/13. \end{aligned} \tag{C.10}$$

Referring to equation C.9: $\xi = [x_1, x_2]'$ where x_1 is motor position and x_2 is motor velocity. The right hand side of equation C.10 represents f. All the states are measured; therefore output function $y(t)$ is equal to states $\xi(t)$ and f is equal to h. According to ACADO syntax, the states to be tracked are determined by their higher weight in matrix Q. This is illustrated by the weighting matrix later in this section. The input voltage v to the motor in equation C.10 is the control u. From previous sections, it was determined that the saturation voltages of the motor are $[-1.76, 1.76]$, these are also taken into account by the controller. The deadzone nonlinearity usually comes into play when implementing the controller. Therefore it is treated later in this chapter. Having related the variables, the controller is rewritten as follows:

$$\underset{\xi(.),\ u(.)}{\text{minimise}} \quad \int_0^T \left(\left\| y(t) - y_{ref}(t) \right\|_Q^2 + \left\| u(t) - u_{ref}(t) \right\|_R^2 \right) dt \tag{C.11a}$$

$$\text{subject to} \quad \xi(0) = 0, \tag{C.11b}$$

$$\forall t \in [0,T]: \quad \dot{\xi}(t) = f\left(t, \xi(t), u(t)\right), \tag{C.11c}$$

$$\forall t \in [0,T]: \qquad y(t) = \xi(t), \tag{C.11d}$$

$$\forall t \in [0,T]: \qquad -1.76 \leq u(t) \leq 1.76. \tag{C.11f}$$

For velocity control the following variables were used in equation C.11; t was in steps of $\frac{1.2}{5}$ seconds, $T = 1.2$ seconds, $Q = \text{diagonal}[1.0\text{e}-8,\ 8.0]$ and $R = [1.0]$. A two-level step in velocity was applied to the controller as reference velocity, and the output velocity of the DC motor was logged and plotted. The reference velocity and actual velocity have been plotted on the same time axis for clarity and ease of interpretation of the results. Figure C.8 shows the graphs of reference velocity and actual motor velocity. Figure C.9 shows the feedback control action plotted against time axis as well.

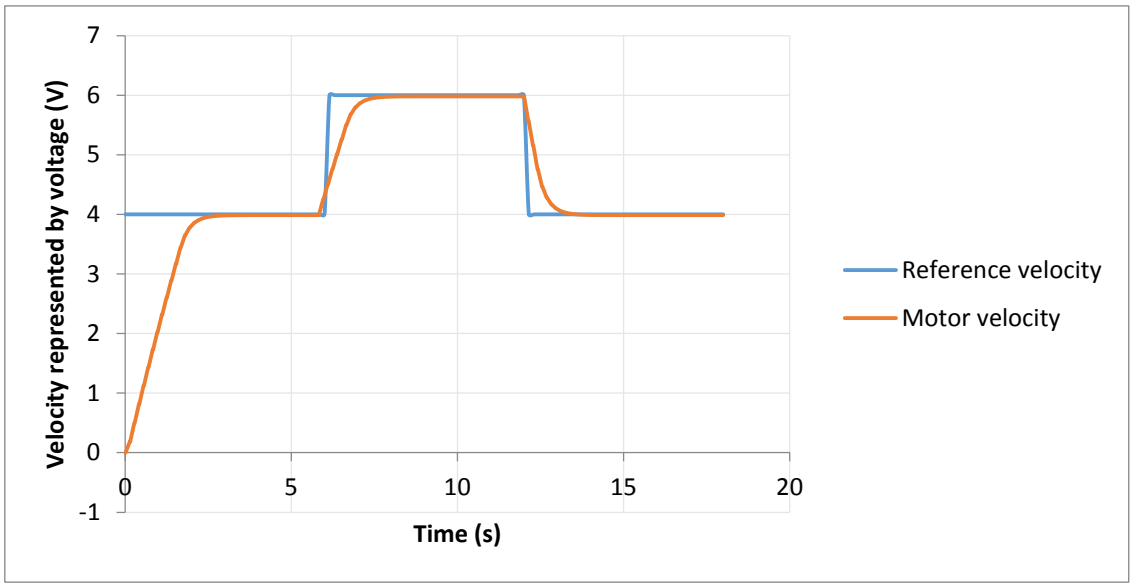

Figure C.8: Showing the DC Motor reference and actual velocity in closed loop simulation. Reference velocity is seen tracked with no steady state error.

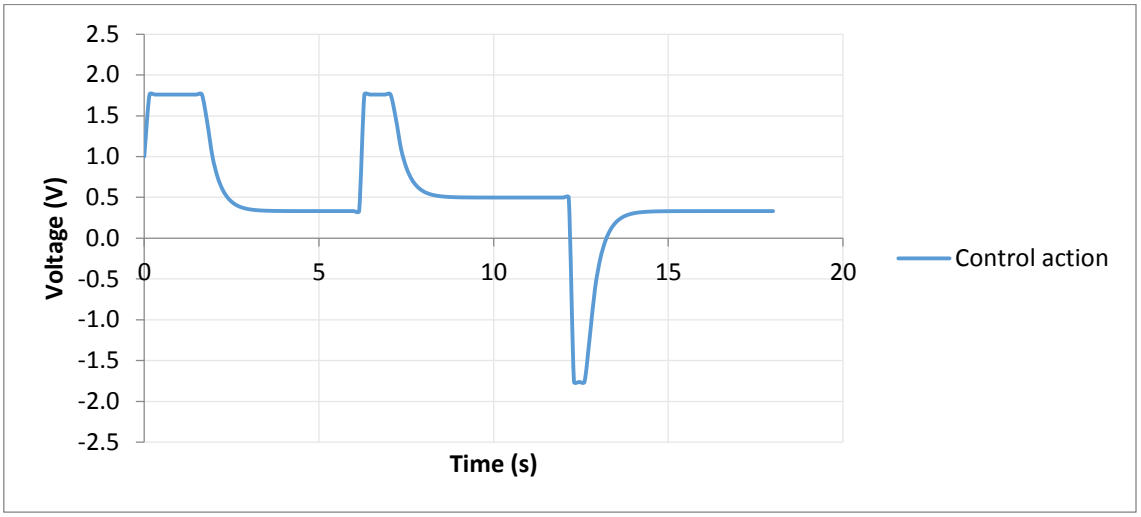

Figure C.9: Showing the DC Motor control action for velocity control in closed loop simulation. Constraint on input voltage is seen respected.

For position control the following variables were used in equation C.11; t was in steps of $\frac{3}{2}$ seconds, $T = 3$ seconds, $Q = \text{diagonal}[5.0,\ 1.0\text{e} - 8]$ and $R = [1.0]$. A two-level step in position was applied to the controller as reference position, and the output position of the DC motor was logged and plotted. The reference position and actual position have been plotted on the same time axis for clarity and ease of interpretation of the results. Figure C.10 shows the graphs of reference position and actual motor position. Figure C.11 shows the feedback control action for position control plotted against time in seconds.

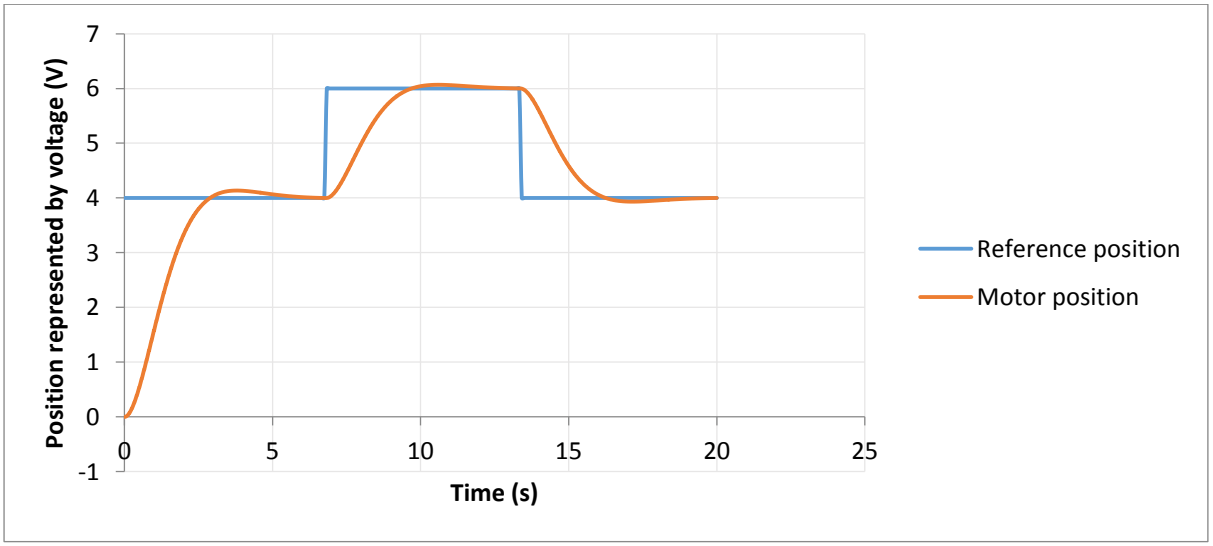

Figure C.10: Showing the DC Motor reference and actual position in closed loop simulation. Reference position is seen tracked with no steady state error

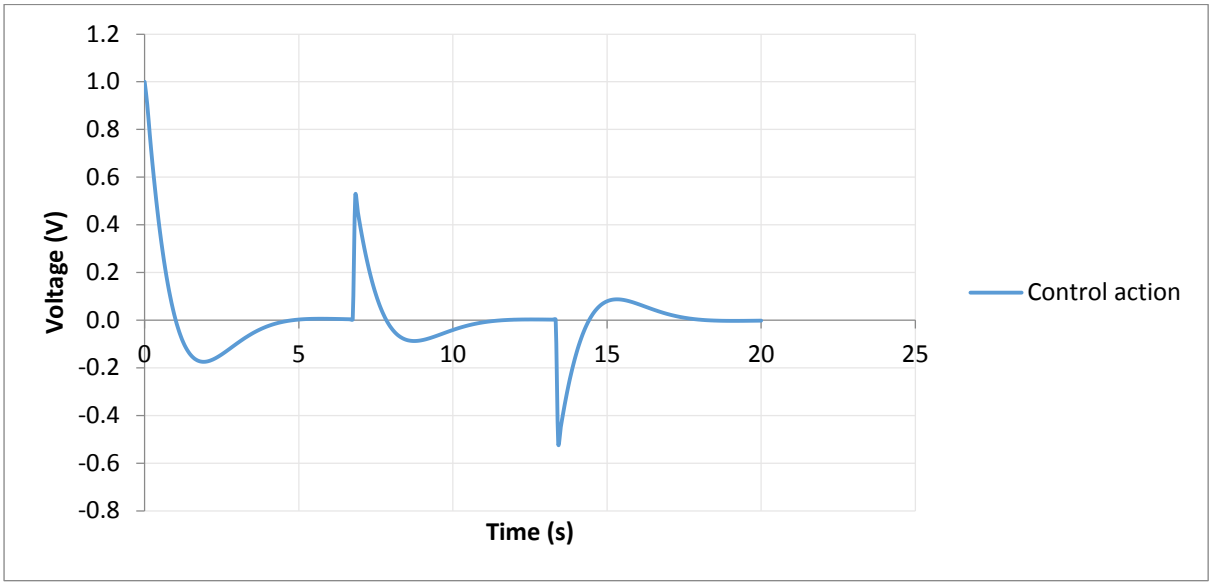

Figure C.11: Showing the DC Motor control action for position control in closed loop simulation. No constraint is violated.

From figures C.8 and C.10, it is observed that the designed controller form is able to track reference velocity and position respectively with zero steady state error. From figure C.9 the controller shows its ability to respect constraints on the control signal, the constraint is due to saturation nonlinearity in DC motor's input voltage. Saturation points were reached three times as shown on the graph C.9; the first two were during acceleration of the motor due to positive step change in reference velocity, the third was due to deceleration (braking) of the motor after negative step change in reference velocity. Saturation constraints were not active during position control as seen in figure C.11.

C.3 DC Motor Implementation Results

After simulation results were satisfactory as observed in the previous section, the next step was implementation of the controller on the physical DC Motor plant. This section is therefore presenting implementation procedure and experimental results.

Both velocity control and position control controllers were implemented in Control and instrumentation laboratory at the University of Cape Town. The implementation was done using ACADO, compiled for Microsoft Visual Studio (MVS) community 2017, running on a windows desktop computer of 4GB RAM and 2.5 GHz Intel Processor. During simulation, it was found that the maximum time taken by the solver to run one simulation (of the simulation environment discussed in chapter 6) was just less than 0.2 seconds. This being the case, the controller sampling time was set to 0.2 seconds, and a function was called in C++ which let the control action written to the physical plant at every 0.2 seconds. This was to make sure the system remains a single rate, rather than multi-rate.

During implementation, it was noted that there were steady state errors in both velocity and position control. Steady state errors are mainly as a result of disturbances and model mismatches in MPC. A solution to handle these errors is to augment the state space model

with either input disturbance or output disturbance as explained in [71]. In this study, the DC motor state space model presented in equation 7.10 was augmented with input disturbance as follows:

$$\begin{gathered} \dot{x}_1 = x_2, \\ \dot{x}_2 = (-x_2 + 12(v))/13 + d_i, \\ \dot{d}_i = d_i. \end{gathered} \tag{C.12}$$

Referring to controller equation C.9, the state was then augmented as $\xi = [x_1, x_2, d_i]'$ where x_1 is motor position, x_2 is motor velocity and d_i is disturbance state. An observer was designed to estimate these three states. The observer designed is of the form:

$$\hat{\dot{\xi}}(t) = f_{aug}\left(t, \xi(t), u(t)\right) + L[(y_m(t) - h_{aug}\left(t, \xi(t), u(t)\right)] \tag{C.13}$$

Where $\hat{\xi}$ represents the estimates of the three states, f_{aug} and h_{aug} are the right hand side of equation C.12 similar to previous explanation under velocity and position simulation section. y_m is measured output and L is the observer gain. The observer gain was designed and the values of the gain used in this study were $L = \text{diagonal}[1.2, 0.7, 1.5]$.

Recalling from previous sections of this chapter, the actual motor velocity is represented as the measured voltage across tachometer in volts. The same scenario applies to position measurement; it is represented as a measured voltage across the output potentiometer in volts. Following this representation, the references for velocity and position are also in volts.

- **Velocity Control Implementation**

For velocity control implementation the following variables were used in equation C.11; t was in steps of $\frac{1.2}{5}$ seconds, $T = 1.2$ seconds, $Q = \text{diagonal}[1.0\text{e}-8,\ 2.5,\ 1.0\text{e}-8]$ and $R = [4.5]$. A two-level step in velocity was applied to the controller as reference velocity, and the output velocity of the DC motor was measured, logged and plotted. The reference velocity and actual velocity are plotted on the same time axis for clarity and ease of interpretation of the results. Figure C.12 shows the graphs of reference velocity and actual motor velocity. Figure C.13 shows the feedback control action plotted against time axis as well.

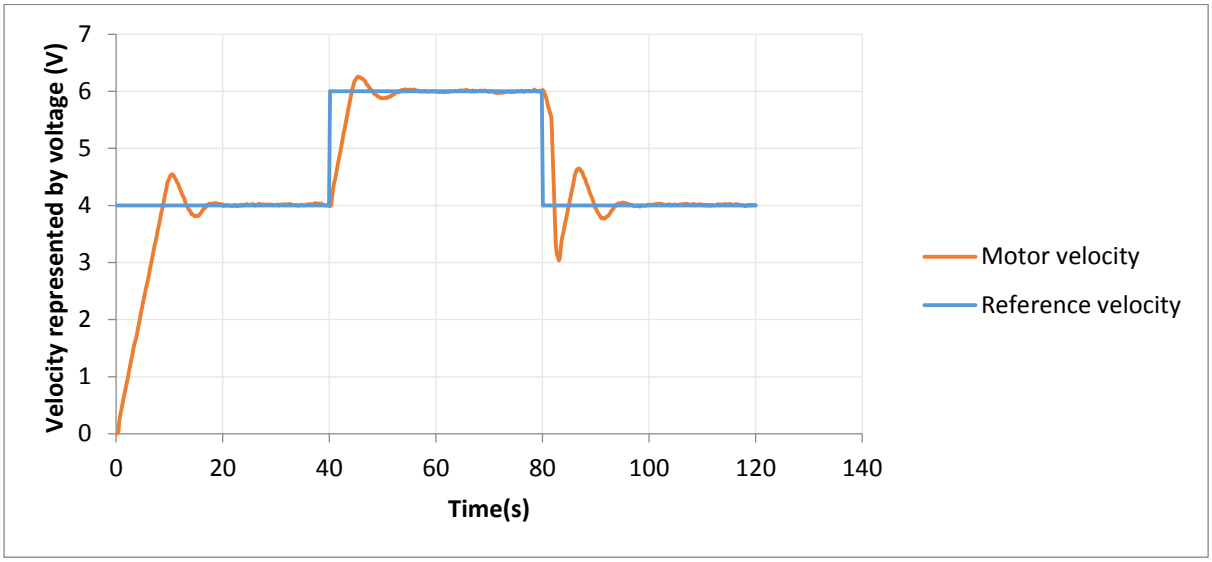

Figure C.12: Showing the DC Motor velocity control implementation results. Reference velocity is seen tracked with no steady state error in implementation.

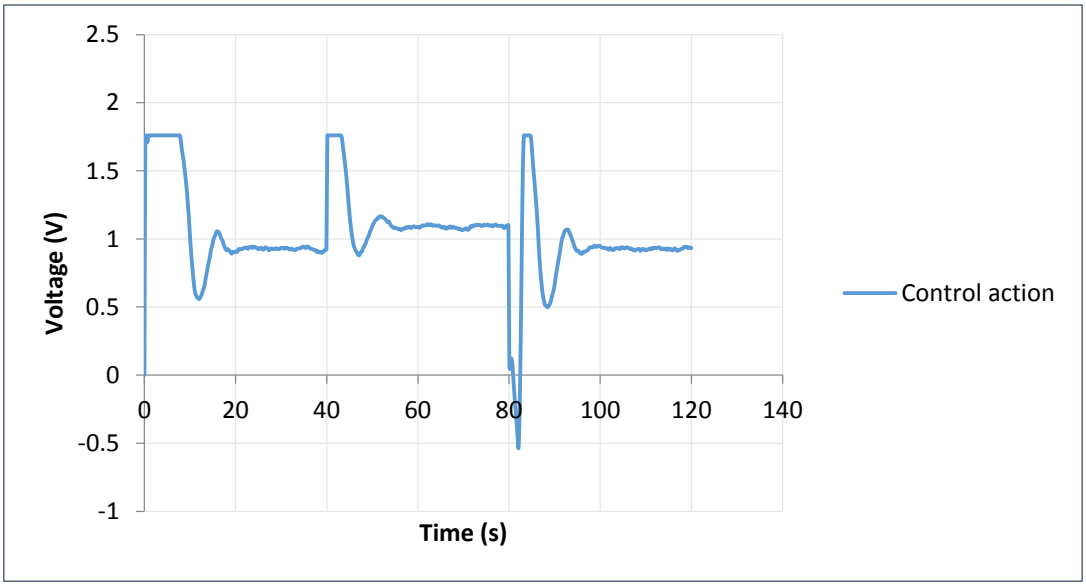

Figure C.13: Showing DC Motor velocity feedback control action. Saturation constraint on input is seen respected in implementation as well.

- **Position Control Implementation**

For position control implementation the following variables were used in equation C.11; t was in steps of $\frac{1.2}{5}$ seconds, $T = 3$ seconds, $Q = \text{diagonal}[5.0, 1.0\text{e}-8,\ 1.0\text{e}-8]$ and $R = [0.01]$.

During position control, the effect of deadzone nonlinearity caused steady state error to persist. This deadzone came into play during small velocities as the motor was about to reach the reference position. The result is that the motor stopped before reaching the actual position, and the calculated control action from the controller was too small to get the motor moving again in trying to reach the actual position. There are a number of ways

of reducing the effect of deadzone. Due to the discontinuous nature of the deadzone, some authors use fuzzy logic and neural networks compensation to identify regions of operations. Therefore, the utilization of fuzzy logic and/or neural network are common as presented by many researchers for example in [72], [73], [74]. Sliding mode control (SMC) is also used frequently to deal with deadzone and other nonlinearities as in [75], [76]. Another option is referred to as adaptive control scheme [77]. In [77] the adaptive deadzone inverse controller was designed which can easily be combined with any of the advanced control methodologies. The deadzone in this study was dealt with by employing similar approach presented in [77].

In simple terms, the approach in this study was implemented as follows: all the calculated control actions that were calculated after the motor had stopped and were within the deadzone determined previously in this chapter, were pushed out of the deadzone proportionally to their value. This pushing out of control actions resulted into some few oscillations about reference position before the motor would actually come to rest. This is evidenced in the results for position control.

A two-level step in position was applied to the controller as reference position, and the output position of the DC motor was measured, logged and plotted. The reference position and actual measured position have been plotted on the same time axis for clarity and ease of interpretation of the results. Figure C.14 shows the graphs of reference position and actual motor measured position. Figure C.15 shows the feedback control action plotted against time axis as well.

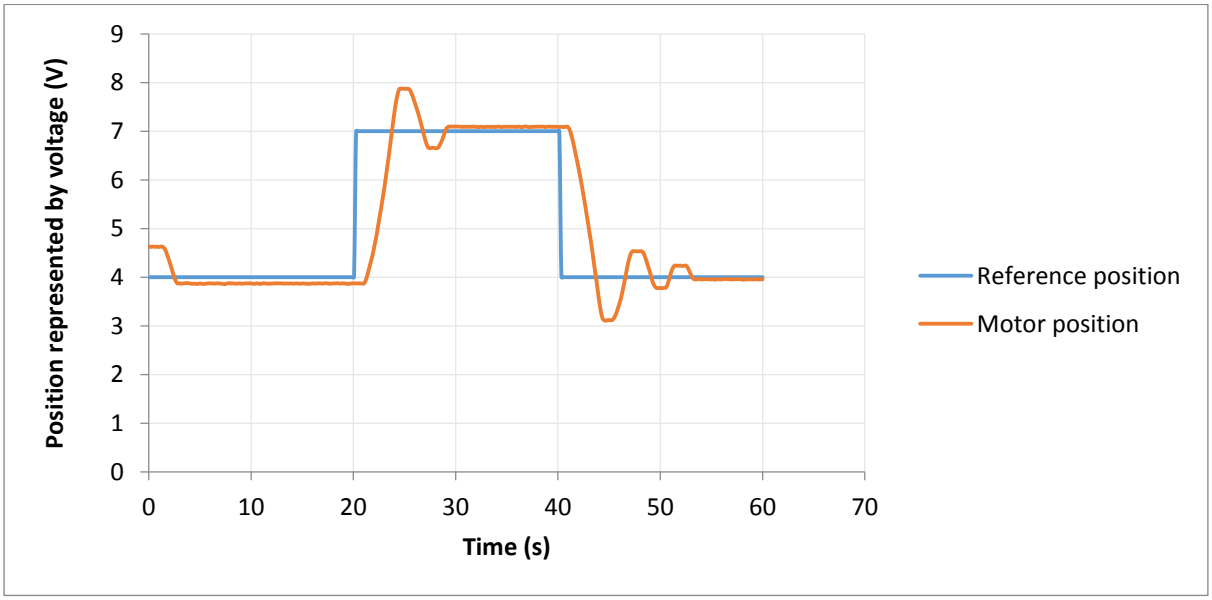

Figure C.14: Showing DC Motor position control implementation results. Reference position is seen tracked with small steady state error of around 3.5%.

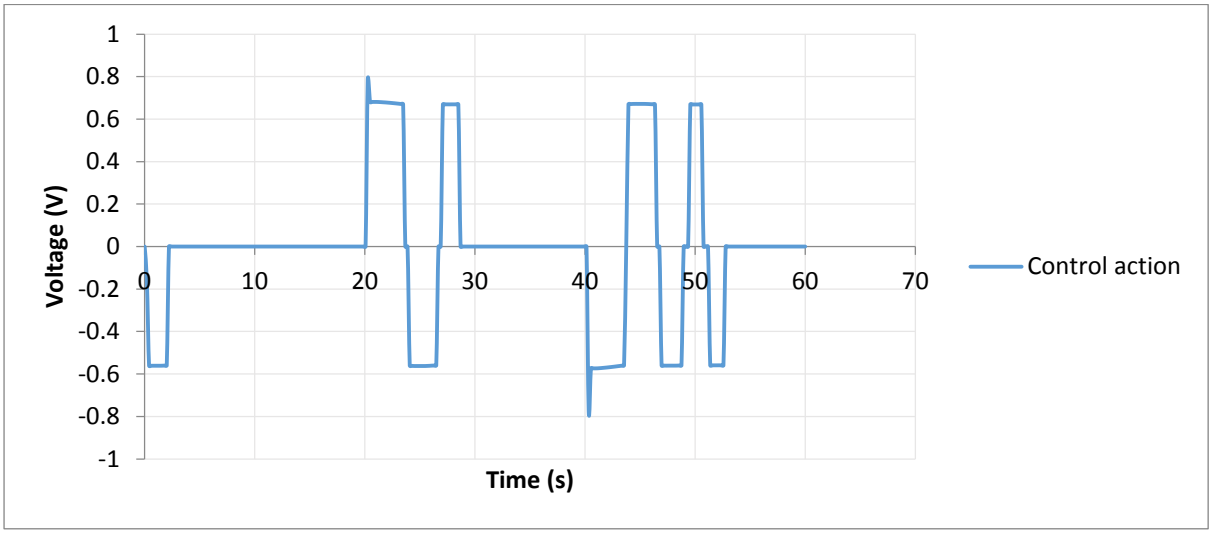

Figure C.15: Showing DC Motor position feedback control action. Saturation was not reached as seen in the figure.

The velocity and position implementation results showed that the designed NMPC controller form works well even in implementation. In velocity control (figure C.12), some oscillations were observed around the reference velocity, which however died within a short period of time. The oscillations were caused by larger values of initial disturbance estimates during step change in reference. These disturbances made the control action to be aggressive hence causing overshoot and undershoot (oscillations) in motor velocity until disturbance estimate converged. Thereafter, the actual velocity tracked the set point with zero stead state error. In addition to set point tracking, it was also observed that the constraint on control action was respected in velocity control. This can be seen from the plot of control action against time graph in figure C.13. In position control (figure C.14), it is also seen that the controller was able to track set point position well, with only around 3.5% maximum steady error. Oscillations observed around set point were as a result of controller pushing out control actions which fell inside deadzone as described in previous section of this chapter. Deadzone was the only nonlinearity that could not be taken care of completely by the controller. The control actions did not reach saturation points to activate constraints in position control as seen in position control action graph in figure C.15.

This appendix has demonstrated the working capability of the proposed controller designed in chapter 6. The controller was simulated and implemented on a DC Motor system. The overall results showed that the designed controller was able to work pretty well in simulation as well as implementation. This gave the confidence that the proposed controller could also work on a higher order system like the nonlinear vehicle model which was the main focus of this study.

Appendix D

D. Resources Used

Some of the source codes used during simulations in the study have been submitted together with this report. The files submitted include the following:

- DC motor position control.
- DC motor velocity control.
- Model-1 simulation file (benchmark model).
- Model-2 simulation file (Robustness testing model).
- Model-6 simulation file (Disturbance rejection model).

To run the codes follow the following simple steps:

i. Install ACADO software by following instructions on the following site: http://acado.github.io/install_linux.html.
ii. Put the C++ file to run in the <ACADO_ROOT>/examples/my_examples folder.
iii. Go to <ACADO_ROOT>/build and type:

```
cmake ..
make
```

iv. Once the build process is over, return to <ACADO_ROOT>/examples/my_examples. The executable file will be there; run the file.

For further information about the source codes used here, please contact either the supervisor or the author on the following emails:

Supervisor: mohohlo.tsoeu@uct.ac.za.
Author: igwayi@poly.ac.mw.